

Algorithms for Distributionally Risk-Receptive and Robust Stochastic Integer Programs and Interdiction Problems

Sumin Kang

Dissertation submitted to the Faculty of
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Industrial and Systems Engineering

Manish Bansal, Chair

Xi Chen

Subhash C. Sarin

Sait Tunc

July 24, 2025

Blacksburg, Virginia

Keywords: Distributionally Risk-Receptive Optimization, Distributionally Robust
Optimization, Interdiction Problem, Stochastic Integer Programming, Decomposition
Method

Copyright 2025, Sumin Kang

Algorithms for Distributionally Risk-Receptive and Robust Stochastic Integer Programs and Interdiction Problems

Sumin Kang

ABSTRACT

This dissertation advances the theory and algorithms of stochastic mixed-integer programs under distributional ambiguity, with a particular focus on adversarial games and interdiction problems. We introduce models within both *distributionally risk-receptive* (DRR) and *distributionally robust optimization* (DRO) frameworks, allowing for an adjustment of decision-maker's risk attitudes in adversarial settings. A key contribution is the study of an optimistic perspective, modeled by the DRR framework, which complements a pessimistic perspective of the conventional DRO paradigm and offers new insights for analyzing adversarial models.

This research considers many algebraic optimization modeling tools such as two-stage and multi-stage stochastic programs, integer and disjunctive programming, decision-independent and decision-dependent uncertainty, and both single- and bi-parameterized recourse problems, in the DRR and DRO frameworks. In particular, the dissertation explores three main directions. In the first direction, we study DRR and DRO two-stage network interdiction problems. For solving them, we propose a new class of cutting planes and develop exact and approximation algorithms with finite convergence results. In the second direction, we extend these frameworks to multi-stage stochastic integer programs with both decision-independent and decision-dependent uncertainty, and introduce solution methods based on cutting plane and reformulation techniques. We further extend these results to multi-stage stochastic disjunctive programs by introducing an extended formulation for their feasible sets. These

models are demonstrated on multi-period interdiction problems for network vulnerability assessment. In the third direction, we study two-stage stochastic integer programs with bi-parameterized recourse, where both the recourse objective and constraints are parameterized by first-stage decisions, under both min-min and min-max structures. These models arise in general interdiction problems, decision-dependent stochastic programs, and DRO problems with decision-dependent ambiguity sets that may be empty for some first-stage decisions. To solve these problems, we propose Lagrangian-integrated L-shaped (L^2) methods, including both exact and approximation variants. Our numerical experiments demonstrate that the proposed methods outperform existing approaches, achieving significant speedups in solving interdiction problem instances and general stochastic programming problem instances. Furthermore, through out-of-sample tests, we show that the DRR framework can effectively identify network vulnerabilities and provide robust solutions under data contamination.

Algorithms for Distributionally Risk-Receptive and Robust Stochastic Integer Programs and Interdiction Problems

Sumin Kang

GENERAL AUDIENCE ABSTRACT

This dissertation advances the study of mathematical optimization tools for decision-making under uncertainty, with a particular focus on problems in adversarial settings, such as assessing infrastructure vulnerabilities under disruption. We model optimization problems with both *risk-receptive* and *risk-averse* decision-makers when the complete information of distributions are unavailable. These modeling frameworks are named *distributionally risk-receptive* (DRR) and *distributionally robust optimization* (DRO) frameworks, respectively. A key contribution is the study of an optimistic perspective, modeled by the DRR framework, which complements a pessimistic perspective of the conventional DRO paradigm and offers new insights for analyzing problems within adversarial environments.

This research addresses a broad range of problem classes. Our settings allow decisions over multiple stages and include discrete or logically disjunctive variables (integer/disjunctive programming). We also consider cases where the realizations of uncertainty are either decision-dependent or decision-independent. For these problem classes, we develop exact and approximate solution methods. Our numerical experiments demonstrate that the proposed methods outperform existing approaches, achieving significant speedups. In terms of practical implications, each chapter of the dissertation explores a distinct direction. In the first direction, we study adversarial problems, namely network interdiction problems, where attacker and defender make decisions sequentially, modeled within the DRR and DRO frameworks. Solving the DRR model, especially, provides a pessimistic viewpoint

of the defender, which can be applied to vulnerability analysis in real-world applications, such as critical infrastructure contingency planning. We show that such insights may be not attainable with conventional stochastic programming models, including DRO models. In the second direction, we extend the DRR and DRO frameworks to multistage decision making, where players act sequentially across multiple periods. These models are applied to multi-period interdiction problems. We demonstrate these models can identify vulnerable network components and determine robust allocation of security resources. We further show that the DRR models are effective in data corruption settings—where input data are adversarially corrupted—outperforming both risk-neutral and DRO models. In the third direction, we study two-stage optimization problems where first-stage decisions affect both the second-stage objective and constraints (i.e., *bi-parameterized recourse*). These models have broad applicability, including general interdiction problems, decision-dependent stochastic programs, and DRO problems with decision-dependent ambiguity sets. Notably, this framework extends conventional stochastic programming by capturing complex settings—common in real-world applications like supply chain management and network security—where early decisions affect not only future costs but also the constraints governing subsequent actions. The bi-parameterized recourse approach thus enhances modeling fidelity for such complex decision-making environments.

Dedication

To Yoonjoo.

Acknowledgments

I would like to express my deepest gratitude to those without whom my research would not have been possible.

- First and foremost, I thank my advisor, Dr. Manish Bansal, for his invaluable guidance, continuous support, and encouragement, which always motivated me to advance my research. My sincere appreciation also goes to my committee members, Dr. Xi Chen, Dr. Subhash C. Sarin, and Dr. Sait Tunc, for their guidance and insightful feedback to my research. I am particularly grateful to Dr. Bansal for his generous support in funding my academic endeavors through the following grants, allowing me to explore this field and to broaden my horizons academically and professionally: National Science Foundation (NSF) Grant CMMI-1824897, Automotive Research Center of the University of Michigan, Ann Arbor, in accordance with Cooperative Agreement W56HZV-19-2-0001, U.S. Army CCDC Ground Vehicle Systems Center (GVSC) Warren, MI, and Commonwealth Cyber Initiative Grants.
- I would like to sincerely thank Dr. Sarin, Dr. Bansal, and Dr. Harsha Nagarajan at Los Alamos National Laboratory for giving an opportunity to do an internship (off-site GRA) funded by NSF Non-Academic Research Internships for Graduate Students (INTERN) Supplement of Grant CMMI-2034503. Also, I am grateful to the ISE department for funding me through graduate teaching assistantships.
- I am grateful to my friends, Seonghun, Yunsoo, Myeongseob, and Hyunwoo, for making my years here enjoyable and memorable, filled with precious memories and adventures. I appreciate our discussions, which contributed to my research and preserved my sanity throughout this journey.

- My heartfelt thanks go to my family, Yoonjoo, Lan, and Ahyeon, whose unwavering support and belief in me enabled me to pursue and achieve my goals.

Contents

List of Figures	xvi
List of Tables	xvii
1 Introduction	1
1.1 SIPs and Interdiction Problems	3
1.1.1 Two-Stage SIPs and Single-Round Interdiction Problems	3
1.1.2 Bi-Parameterized Two-Stage Stochastic Programs	6
1.1.3 Multistage SIPs and Multi-Round Interdiction Problems	7
1.2 Distributionally Risk-Receptive and Robust Optimization Models	8
1.3 Contributions and Organization of Dissertation	9
2 Literature Review	12
2.1 Literature Review	12
2.1.1 Two-Stage Stochastic Integer Programs	12
2.1.2 Network Interdiction Problems	14
2.1.3 Bi-Parameterized Two-Stage Stochastic Programs	17
2.1.4 Multistage Stochastic Integer Programs	18
2.1.5 Distributionally Risk-Receptive Optimization	21

3	Distributionally Risk-Receptive and Robust Network Interdiction Problems with General Ambiguity Set	23
3.1	Introduction	23
3.2	Problem Definition: DRO-SPIP and DRR-SPIP	24
3.2.1	Illustrative Example	27
3.3	Solution Approaches for DRR-SPIP	29
3.3.1	Reformulation-Based Approach for DRR-SPIP	30
3.3.2	Cuts-Based Decomposition Approach for DRR-SPIP	31
3.3.3	Restrict-and-Append Algorithm	34
3.4	Exact Solution Approaches for DRO-SPIP	35
3.4.1	Decomposition Approach for DRO-SPIP	36
3.4.2	Relax-and-Append Algorithm for DRO-SPIP	38
3.5	Computational Results	39
3.5.1	Instance Generation and Experiment Setup	41
3.5.2	Computational Results for DRR-SPIP	42
3.5.3	Computational Results for DRO-SPIP	45
3.5.4	Comparison between Risk-Neutral, Distributionally Robust, and Distributionally Risk-Receptive Solutions	49
3.6	Conclusion	52
4	Distributionally Risk-Receptive and Robust Multistage Stochastic Integer Pro-	

grams and Interdiction Models	53
4.1 Introduction	53
4.1.1 Problem Formulation: DRO-MSP and DRR-MSP	55
4.2 Convex Approximations for DRR-MSIPs having Finite Supports	61
4.2.1 A Cutting Plane-Based Approximation for DRR-MSIP	62
4.2.2 A Reformulation-Based Approximation for DRR-MSIP	63
4.3 Convex Approximation for DRR-MSIPs having Continuous Supports and Decision-Dependent Ambiguity Sets	64
4.4 Convex Approximations for DRO-MSIPs	67
4.4.1 A Cutting Plane-Based Approximation for DRO-MSIP	67
4.4.2 A Reformulation-Based Approximation for DRO-MSIP	68
4.5 SDDP-Based Algorithms for DRR-MSP and DRO-MSP	70
4.5.1 Distributionally Ambiguous SDDP	70
4.5.2 Finite Convergence	74
4.6 Extensions to Multistage Stochastic Disjunctive Programs with Distributional Ambiguity	75
4.6.1 DA-SDDP Algorithms for DRR- and DRO-MSDPs	75
4.6.2 Application of Proposition 4.7 for Solving DRR- and DRO-MSIPs using Hierarchical Relaxations	77
4.7 Computational Tests	79

4.7.1	MS-MFIP with Distributional Ambiguity	80
4.7.2	MS-FLIP with Distributional Ambiguity	89
4.8	Conclusion	91
5	Bi-Parameterized Two-Stage Stochastic Min-Max and Min-Min Mixed Integer Programs	94
5.1	Introduction	94
5.1.1	Applications of Bi-Parameterized Stochastic Programs	97
5.1.2	Contributions and Organization	100
5.2	An Exact Algorithm for the Min-Max BTSP	101
5.2.1	A Lagrangian Dual Formulation	101
5.2.2	The Lagrangian-Integrated L-Shaped Method	105
5.3	An Exact Algorithm for the Min-Min BTSP	106
5.4	A Regularization-Augmented Algorithm for BTSPs	110
5.5	Distributionally Robust BTSPs with Finite and Continuous Support	114
5.6	BTSP-Based Reformulations for Stochastic Optimization and Interdiction Problems	118
5.6.1	Risk-Averse Stochastic Optimization with Decision-Dependent Probability Distribution	118
5.6.2	Two-Stage DRO with Decision-Dependent Ambiguity Set	120
5.6.3	Bi-Parameterized Stochastic Network Interdiction Problem	122

5.7	Computational Results	124
5.7.1	Bi-Parameterized (Min-Min) Facility Location Problem	124
5.7.2	Bi-Parameterized (Min-Max) Network Interdiction Problem	129
5.7.3	Distributionally Robust Facility Location with Decision-Dependent Ambiguity Set	133
5.8	Conclusion	139
6	Conclusion	140
	Bibliography	143
	Appendices	158
	Appendix A Appendix for Chapter 3	159
A.1	Proofs	159
A.1.1	Proof of Theorem 3.1	159
A.1.2	Proof of Theorem 3.2	161
A.1.3	Proof of Theorem 3.3	162
A.1.4	Proof of Theorem 3.4	162
A.2	L-Shaped Method with Branch-and-Cut for DRR-SPIP	163
	Appendix B Appendix for Chapter 4	165
B.1	Formulations of the Network Interdiction Problems	165

B.1.1	Formulation of MS-MFIP	165
B.1.2	Formulation of MS-FLIP	167
B.2	Proofs	169
B.2.1	Proof of Theorem 4.1	169
B.2.2	Proof of Proposition 4.2	170
B.2.3	Proof of Theorem 4.3	170
B.2.4	Proof of Theorem 4.4	173
B.2.5	Proof of Theorem 4.5	175
B.2.6	Proof of Theorem 4.6	175
B.2.7	Proof of Proposition 4.7	176
B.2.8	Proof of Proposition 4.8	177
B.3	Pseudocode of DA-SDDP-DP Algorithm	178
Appendix C Appendix for Chapter 5		180
C.1	Proofs	180
C.1.1	Proof of Theorem 5.1	180
C.1.2	Proof of Lemma 5.2	181
C.1.3	Proof of Theorem 5.3	182
C.1.4	Proof of Proposition 5.4	182
C.1.5	Proof of Proposition 5.5	183

C.1.6 Proof of Proposition 5.6	184
--	-----

List of Figures

3.1	Example network with source node $s = 1$ and destination node $t = 4$	27
3.2	Optimal objective values for instances of N40, US100, obtained by solving DRO-SPIP (DRA), DRR-SPIP (DRR), and S-SPIP (RN) with Wasserstein ambiguity set and $b \in \{2, 5, 10\}$	50
3.3	Optimal objective values for instances of N60, US100, obtained by solving DRO-SPIP (DRA), DRR-SPIP (DRR), and S-SPIP (RN) with moment matching set defined using $\epsilon_M \in \{0.05, 0.1, 0.25\}$	50
3.4	Optimal objective values for instances of N80, US100, obtained by solving DRO-SPIP (DRA), DRR-SPIP (DRR), and S-SPIP (RN) with Wasserstein ambiguity set defined using $\rho \in \{0.1, 0.2, 0.5\}$	51
4.1	Plots of percentiles from out-of-sample tests on DRO-MFIP (90th and 95th percentiles) and DRR-MFIP (5th and 10th percentiles)	84
4.2	A network in a 5×2 grid shape.	86
4.3	Comparison of interdiction solutions from different models in out-of-sample test. Network has a 5×2 grid shape, and contamination level $\bar{\alpha}$ is 0.4.	88
4.4	Comparison of interdiction solutions from different models in out-of-sample test. Network has a 6×3 grid shape, interdiction budget b is 5, and contamination level $\bar{\alpha}$ is 0.4.	89
5.1	An example of a nonconvex expected recourse function.	96

List of Tables

3.1	Expected travel cost for a given interdicator's action $x \in X = \{x^1, \dots, x^6\}$ and probability distribution $P \in \{P_1, P_2, P_3\}$; optimal solution and objective value of S-SPIP for each distribution P , DRR-SPIP, and DRO-SPIP.	28
3.2	Computational results for solving DRR-SPIP US20 instances using reformulation-based algorithm, cut-based decomposition algorithm, and restrict-and-append algorithm.	43
3.3	Computational results for solving DRR-SPIP US100 instances using reformulation-based algorithm, cut-based decomposition algorithm, and restrict-and-append algorithm.	44
3.4	Computational results for solving DRO-SPIP US20 instances using decomposition algorithm and relax-and-append algorithm.	46
3.5	Computational results for solving DRO-SPIP US100 instances using decomposition algorithm and relax-and-append algorithm.	47
4.1	Details of DRR- and DRO-MFIP instances	81
4.2	Performance comparison of algorithms for DRR- and DRO-MFIP instances .	82
4.3	Optimal interdiction solutions for the DRR and DRO models with $\bar{\alpha} \in \{0.2, 0.4, 0.6\}$ and $\epsilon \in \{0, 10, 20, 40\}$	87
4.4	Details of DRR- and DRO-FLIP instances	90
4.5	Performance comparison of algorithms for DRR- and DRO-FLIP instances .	92

5.1	Results for BiFLP instances.	128
5.2	Results for DR-BiFLP instances.	130
5.3	Results for BiNIP instances.	132
5.4	Results from L^2 -R and DA for the DRFLP instances.	137
5.5	Results from L^2 and L^2 -R for DRFLP instances.	138

List of Abbreviations

BiFLP: Bi-parameterized Facility Location Problem

BiNIP: Bi-parameterized Network Interdiction Problem

BTSP: Bi-parameterized Two-stage Stochastic Program

CVaR: Conditional Value-at-Risk

DA-SDDP: Our customized SDDP that addresses Distributional Ambiguity

DRFLP: Distributionally Robust Facility Location Problem

DRO-SDDP-C: DA-SDDP for DRO-MSIPs with Cutting plane-based approximation ([4.22](#))

DRO-SDDP-R: DA-SDDP for DRO-MSIPs with Reformulation-based approximation ([4.23](#))

DRO: Distributionally Robust Optimization

DRR-SDDP-C: DA-SDDP for DRR-MSIPs with Cutting plane-based approximation ([4.10](#))

DRR-SDDP-R: DA-SDDP for DRR-MSIPs with Reformulation-based approximation ([4.13](#))

DRR: Distributionally Risk-Receptive / Distributionally Risk-Receptiveness

L² Method: Lagrangian-integrated L-shaped Method

LP: Linear Programming

MIP: Mixed-integer Programming

MS-FLIP: Multistage Stochastic Facility Location Interdiction Problem

MS-MFIP: Multistage Stochastic Maximum Flow Interdiction Problem

MSDP: Multistage Stochastic Disjunctive Program

MSIP: Multistage Stochastic Integer Program

MSLP: Multistage Stochastic Linear Program

NIP: Network Interdiction Problem

SDDP: Stochastic Dual Dynamic Programming

SIP: Stochastic Integer Program

SPIP: Shortest Path Interdiction Problem

Chapter 1.

Introduction

Stochastic programming is a framework for modeling optimization problems under uncertainty, typically assuming that the *true* probability distribution of uncertain parameters is known or can be approximated using prior knowledge or historical data. In practice, however, such distributional information is often unavailable or limited, which can result in poor approximations and, consequently, poor solution quality. Distributionally robust optimization (DRO) addresses this challenge by enabling a decision-maker to hedge against such *distributional ambiguity* [89]. Specifically, a DRO model defines an *ambiguity set*, consisting of potential distributions, and seeks an optimal solution against worst-case distributions within this set. Despite successful applications of DRO, it can be insufficient for problems involving multiple decision-makers when it is necessary to account for their distinct and potentially conflicting perspectives.

For example, consider a Stackelberg game involving two players: an attacker and a defender. The attacker acts as the leader, selecting their strategy first, while the defender, as the follower, observes the attacker's choice and then responds. The attacker's goal is to maximally degrade the defender's performance. Adversarial games of this type are commonly used to evaluate vulnerabilities within the defender's system through a hypothetical adversary (see, e.g., [20, 26, 49, 93]). Naturally, uncertainty in the attacker's decisions and their impacts can significantly affect the defender's analysis, thereby highlighting the importance of addressing distributional ambiguity.

However, applying DRO in this context implicitly models an optimistic perspective for the defender by assigning lower weights to successful attack scenarios, since these are less likely under the attacker’s worst-case distributions. In other words, while DRO can effectively capture the attacker’s pessimistic viewpoint, it fails to reflect the defender’s pessimistic perspective.

To address this issue, we introduce a new framework, termed *distributionally risk-receptive* (DRR) optimization, where a decision-maker is receptive to distributional ambiguity, in contrast to the conventional DRO approach. In DRR, optimal solutions are determined with respect to best-case distributions within the ambiguity set, explicitly modeling the decision-maker’s optimistic viewpoint. In the example above, this approach adopts an optimistic perspective for the attacker, which corresponds to a pessimistic perspective for the defender. Throughout this dissertation, we use the term *(distributionally) risk-averse* to describe the risk attitude of decision-makers in DRO models, thereby clearly distinguishing it from and juxtaposing it with the *(distributionally) risk-receptive* attitude. It is worth noting that concepts related to DRR have also been explored in literature on optimization under data contamination, particularly in robust statistics [17, 18, 48] and Rockafellian relaxations [84]; we discuss these topics further in Section 2.1.

In this dissertation, we study stochastic integer programs (SIPs) within the DRR and DRO frameworks, with a particular focus on interdiction problems—a special class of attacker-defender games. Specifically, we investigate single-round interdiction problems as two-stage stochastic programs and multi-round interdiction problems as multistage stochastic programs (MSPs). These problems are characterized by their min-max structures, discrete decision variables, and uncertainty, which together present significant challenges for efficient solution methods. Furthermore, we extend the classical SIP formulation to a *bi-parameterized* variant, thereby addressing a broader class of problems in DRO and interdiction. We develop and

present solution approaches for these formulations, leveraging decomposition methods that employ cutting planes and reformulation techniques.

The remainder of this chapter is organized as follows. We begin by providing further details on the formulations of two-stage and multistage SIPs and interdiction problems, and then present their extensions within the DRR and DRO frameworks. Next, we introduce the bi-parameterized formulation of SIPs. Finally, we present an overview of the dissertation, summarizing the main contributions of each chapter.

Below, we present some of the key notation used throughout this dissertation.

Notation. For any positive integer d , we define $[d]$ as $1, \dots, d$. The convex hull of a set X is denoted by $\text{conv}(X)$. Given a set X and a subspace Y , $\text{Proj}_Y(X)$ denotes the projection of X onto Y . The expectation with respect to a probability measure P is written as $\mathbb{E}_P[\cdot]$; when the probability measure is clear from context, we omit the subscript. The sets of real numbers and nonnegative real numbers are denoted by \mathbb{R} and \mathbb{R}_+ , respectively; also, the sets of integers and nonnegative integers are denoted by \mathbb{Z} and \mathbb{Z}_+ .

1.1 SIPs and Interdiction Problems

1.1.1 Two-Stage SIPs and Single-Round Interdiction Problems

Let $\boldsymbol{\xi}$ be a random vector that represents uncertain parameters of our interest. In two-stage stochastic programs, first-stage decisions are made before observing a realization of $\boldsymbol{\xi}$, and second-stage (*recourse*) decisions are made afterward. A standard formulation of two-stage

SIPs is given by

$$\min c^\top x + \mathbb{E}[Q(x, \omega)] \quad (1.1a)$$

$$\text{s.t. } Ax \geq b \quad (1.1b)$$

$$x \in \mathcal{X} \quad (1.1c)$$

We define the random vector as $\boldsymbol{\xi} = (q, r, W, T)$. For each scenario $\omega \in \Omega$, its realization is denoted by $\boldsymbol{\xi}(\omega) = (q(\omega), r(\omega), W(\omega), T(\omega))$, and the value function $Q(x, \omega)$, so-called *recourse function*, is defined as:

$$Q(x, \omega) := \min q(\omega)^\top y \quad (1.2a)$$

$$\text{s.t. } W(\omega)y \geq r(\omega) - T(\omega)x \quad (1.2b)$$

$$y \in \mathcal{Y}. \quad (1.2c)$$

Here, the sets $\mathcal{X} \subseteq \mathbb{R}_+^{d_x}$ and $\mathcal{Y} \subseteq \mathbb{R}_+^{d_y}$ impose integrality restrictions on decision variables x and y , respectively.

Network interdiction problems (NIPs) are a special class of attacker-defender games that can be modeled as integer programs with two stages and conflicting (min-max) objectives. In this context, the leader is referred to as the *interdictor* and the follower is referred to as the *network user*. The interdictor selects a subset of network components to disrupt, aiming to maximally degrade network performance; then, the network user responds by optimizing their objective on the disrupted network. A standard formulation of single-round stochastic NIPs is given by (1.1), where $\boldsymbol{\xi} := (q, r, G, W, T)$, and the recourse function $Q(x, \omega)$ for each

scenario $\omega \in \Omega$ is defined as follows:

$$Q(x, \omega) = \max (q(\omega) + G(\omega)^\top x)^\top y \quad (1.3a)$$

$$\text{s.t. } W(\omega)y \geq r(\omega) - T(\omega)x \quad (1.3b)$$

$$y \in \mathcal{Y}. \quad (1.3c)$$

Note that the second-stage (network user's) problem is a maximization problem, representing the adversarial relationship between the interdicator and the network user. An example of NIPs is the shortest path interdiction problem [26, 46], where disruptions penalize the use of attacked arcs; see formulation (3.1) in Chapter 3 for more details.

A common solution method for two-stage SIPS and NIPs involves approximating the value function $Q(x, \omega)$ using valid cutting planes—called *optimality cuts*—of the form of

$$Q(x, \omega) \geq \alpha^\top x + \beta.$$

These cuts collectively form a piecewise linear and convex approximation of the value function, i.e., $Q(x, \omega) \geq \max_{(\alpha, \beta)} \{\alpha x + \beta\}$. By using this convex approximation as a proxy of Q , we can obtain efficient solutions to the interdicator's problem. But, the effectiveness of this technique relies on the quality of these cuts; ideally, these cuts should be *tight* to the value function, meaning that they satisfy $Q(\hat{x}, \omega) = \alpha^\top \hat{x} + \beta$ for some feasible point \hat{x} . Another common method for NIPs is to take a dual of the inner maximization and solve the whole problem as a single-level minimization problem; for example, if the inner problem is a linear program, i.e., $\mathcal{Y} = \mathbb{R}_+^{d_y}$, we can take the linear programming (LP) dual of the inner problem. In general, the resulting dual problem involves bilinear terms of the first-stage variables and second-stage dual variables, which may pose computational challenges.

1.1.2 Bi-Parameterized Two-Stage Stochastic Programs

A formulation of bi-parameterized two-stage stochastic programs (BTSPs) is given as follows:

$$\min \left\{ c^\top x + \mathbb{E}[Q(x, \omega)] : Ax \geq b, x \in \mathcal{X} \right\},$$

where the bi-parameterized recourse function $Q(x, \omega)$ for each $\omega \in \Omega$ is defined as

$$Q(x, \omega) = \min (q(\omega) + G(\omega)^\top x)^\top y \tag{1.4a}$$

$$\text{s.t. } W(\omega)y \geq r(\omega) - T(\omega)x \tag{1.4b}$$

$$y \in \mathcal{Y}. \tag{1.4c}$$

Here, the recourse problem has both objective function and constraints, impacted by first-stage decisions x , in contrast to the *single-parameterized* recourse (1.2) in the standard formulation of two-stage SIPs. The bi-parameterization can enrich the applicability of SIPs by enabling the modeling of first-stage decision's impacts on recourse costs. Furthermore, as shown in Chapter 5, bi-parameterized stochastic programs can be used to model decision-dependent uncertainty in stochastic programming and DRO models.

However, although the bi-parameterization may appear to be a straightforward extension, it introduces significant challenges for solution methods. These challenges are primarily derived from the nonconvexity and discontinuity of the recourse functions, derived by the bi-parameterization and the presence of discrete variables.

We also note that the bi-parameterized setting in NIPs remains largely unexplored, despite the standard form (1.3); most NIP studies focus on single-parameterized problems—i.e., assuming either $G(\omega) = 0$ (e.g., [26, 47, 64, 85]) or $T(\omega) = 0$ (e.g., [46, 72, 95])—for efficient

solution methods. This dissertation bridges this gap by addressing BTSPs in both min-min (i.e., SIPs) and min-max (i.e., NIPs) structures in Chapter 5.

1.1.3 Multistage SIPs and Multi-Round Interdiction Problems

We now extend SIPs to multistage stochastic programming and their applications to multi-round NIPs. Let x_t be the decision vector for stage $t \in [T]$, where $[T] := \{1, \dots, T\}$. An MSP can be formulated as

$$\min_{x_1 \in X_1} \left\{ f_1(x_1) + \mathbb{E}_{P_2} [Q_2(x_1, \omega_2)] \right\}, \quad (1.5)$$

where $\omega_t \in \Omega_t$ is a scenario associated with uncertain data in each stage $t = 2, \dots, T$, and the *cost-to-go functions* are defined as

$$Q_t(x_{t-1}, \omega_t) = \min_{x_t \in X_t(x_{t-1}, \omega_t)} \left\{ f_t(x_t, \omega_t) + \mathbb{E}_{P_{t+1}} [Q_{t+1}(x_t, \omega_{t+1})] \right\}, \quad (1.6)$$

for $t = 2, \dots, T$, and $Q_{T+1} = 0$. Set X_1 denotes the feasible region of the first stage, and set $X_t(x_{t-1}, \omega_t)$ denotes the feasible region of each stage $t \in \{2, \dots, T\}$ that depends on a decision of the previous stage x_{t-1} . Given scenario ω_t , the random vector ξ_t at stage t is realized as $\xi_t(\omega_t)$ with probability $P_t(\omega_t)$. When sets X_t are defined by mixed-integer linear sets, we refer to the corresponding problem (1.5) as multistage stochastic integer program (MSIP).

Multi-round NIPs can be formulated as (1.5) with the functions f_2, \dots, f_T are defined as value functions of the network user's problems:

$$f_t(x_t, \omega_t) = \max_{y_t \in Y_t(x_t, \omega_t)} g_t(x_t, y_t, \omega_t) \quad \text{for } t = 2, \dots, T, \quad \omega_t \in \Omega_t. \quad (1.7)$$

Notice that the problem at each stage t involves an outer minimization and an inner maximization, i.e., a min-max structure.

Solution methods for two-stage SIPs and single-round NIPs—such as those based on optimality cuts and dualization—can be adapted to address each stage’s problem in the multistage setting. However, a critical challenge is to effectively approximate each stage’s optimal solution, while accounting for all the subsequent stages and their associated costs. In Chapter 4, we present solution methods, addressing this challenge, illustrated with examples including multistage stochastic maximum flow interdiction and facility location interdiction problems.

1.2 Distributionally Risk-Receptive and Robust Optimization Models

We now consider the case where the true probability distributions are unknown or cannot be reliably approximated. Recall that we model the decision-maker’s response to this ambiguity using two frameworks: DRO and DRR. Within these frameworks, risk preferences are captured by the decision-maker’s focus on distributions within the ambiguity sets: worst-case distributions for DRO models and best-case distributions for DRR models. Ambiguity sets can be constructed using discrepancy metrics [10, 63, 80], moment information [27], and other approaches (see [81] for additional examples). We will introduce specific ambiguity sets relevant to our models as needed in the following chapters.

A standard formulation of DRO two-stage stochastic programs is given by

$$\min_{x \in X} c^\top x + \max_{P \in \mathcal{P}} \mathbb{E}_P[Q(x, \omega)] \quad (1.8)$$

where Q is defined as in (1.2) or (1.3), and \mathcal{P} is the ambiguity set.

In contrast, a standard formulation of DRR two-stage stochastic programs is given by

$$\min_{x \in X} c^\top x + \min_{P \in \mathcal{P}} \mathbb{E}_P[Q(x, \omega)]. \quad (1.9)$$

Here, the inner minimization over \mathcal{P} serves to identify the best-case distributions for the first-stage decision-maker.

DRR models may appear to have a simple structure, but they face significant computational challenges. Directly solving a DRR model as a single-level minimization is impractical due to the integral defining the expectation, unless the ambiguity set is specifically chosen. When the support of ξ is finite, the model can be simplified, as the expectation is expressed as a summation. However, this transforms the model into a mixed-integer nonconvex program, which is notoriously difficult to solve. Even without any integrality restrictions, the objective function of a DRR model is nonconvex. While DRO models have been extensively studied, e.g., [5, 6, 31, 32, 35, 80, 106], very few studies have investigated solution approaches for DRR models. To the best of our knowledge, this research is the first to explore SIPs and NIPs in the DRR framework.

1.3 Contributions and Organization of Dissertation

This dissertation advances the theory and computational methods for stochastic integer programs in the DRR and DRO frameworks, with a particular focus on interdiction problems—and more generally, min-max formulations. We address a broad range of settings, including two-stage/multistage optimization, integer/disjunctive programming, and decision-independent/decision-dependent uncertainty. The specific contributions of each chapter are

outlined below.

- **Chapter 2.** This chapter investigates single-round NIP models in the DRO and DRR frameworks, thereby generalizing classical deterministic and stochastic NIPs. These models enable the analysis of NIPs under varying risk preferences toward distributional ambiguity—distributionally risk-averse and risk-receptive. To solve these models, we develop exact and approximation algorithms, utilizing a new class of cutting planes and reformulation techniques. We present a family of ambiguity sets and provide conditions under which the algorithms are guaranteed to converge in finitely many iterations. Through numerical experiments, we demonstrate that DRO solutions provide robust resource allocations for the interdictor, and that DRR solutions can identify the network user’s critical components in the network. The results also show that the cutting plane-based algorithms are faster than the reformulation-based algorithms, with speedups ranging from 34 times to 395 times across various DRR problem instances. The work presented in this chapter has been published in [49].
- **Chapter 3.** This chapter studies multistage DRR and DRO stochastic mixed-integer programs (DRR- and DRO-MSIPs) and applies these frameworks to multi-round NIPs. We present cutting plane-based and reformulation-based algorithms, and these are able to handle both models with decision-independent and decision-dependent uncertainty. We prove that these methods converge to optimal decisions in finitely many iterations with probability one. Moreover, we introduce generalizations of DRR- and DRO-MSIPs to disjunctive programming, along with solution algorithms. Numerical experiments were conducted on two multi-round NIPs with distributional ambiguity. The results show that the cutting plane-based approaches are on average 28 times and 24 times faster than the reformulation-based approaches for DRO and DRR problem instances, respectively. Out-of-sample tests show the significance of the DRR framework in identifying network

vulnerabilities and mitigating the impact of data corruption. The work presented in this chapter has been published in [51].

- **Chapter 4.** This chapter focuses on BTSPs, addressing both min-min and min-max formulations with mixed-integer variables. Their applications include decision-dependent DRO, decision-dependent (risk-averse) stochastic programs, and bi-parameterized NIPs (as in the standard form (1.3)). To solve BTSPs, we propose Lagrangian-integrated L-shaped (L^2) methods, where a Lagrangian relaxation is used to derive a single-parameterization reformulation, and then a Benders decomposition approach is applied. The L^2 method becomes exact when the first-stage variables are binary. For mixed-binary first-stage variables, we propose a regularization-augmented L^2 method. We also extend these methods to DRO problems with both finite and continuous supports. Numerical experiments demonstrate the efficiency of the L^2 methods for bi-parameterized NIP instances, outperforming benchmark methods. For DRO instances with decision-dependent ambiguity sets, the L^2 method achieves the optimal solution on average 5.3 times faster than existing methods.

It is important to note that the L^2 method can address decision-dependent DRO models, even when the ambiguity sets are not guaranteed to be nonempty for all feasible first-stage decisions—which has been a limitation of existing methods for these models.

We refer readers to the *List of Abbreviations* (page xx) for the major abbreviations used throughout this dissertation. For convenience, all proofs are provided separately in the appendix corresponding to each chapter.

Chapter 2.

Literature Review

2.1 Literature Review

This chapter reviews the relevant literature on stochastic integer programs (SIPs), network interdiction problems (NIPs) and distributionally risk-receptive optimization (DRR) models. We focus on foundational models and recent advances in solution methods, particularly those most relevant to the contributions of this dissertation.

2.1.1 Two-Stage Stochastic Integer Programs

Two-stage SIPs have been applied to a wide range of practical problems, such as healthcare [105], power systems [109, 110], airline crew scheduling [104], wildfire planning [71], network interdiction [93], and many more [61, 92, 97].

Although there is extensive literature on two-stage SIPs, the following review focuses specifically on solution methods relevant to the contributions of this dissertation.

The main paradigm for solving two-stage stochastic programs is Benders decomposition [14], also known in the stochastic programming literature as the L-shaped method [99]. The key idea is to construct a convex approximation of the recourse function Q by using optimality cuts, which are essentially valid inequalities for the epigraph $\{(x, \theta) \in X \times \mathbb{R} : \theta \geq \mathbb{E}[Q(x, \omega)]\}$. In some methods [16, 96], the epigraph of $Q(x, \omega) = \{(x, \theta(\omega)) \in X \times \mathbb{R} :$

$\theta(\omega) \geq Q(x, \omega)$ is approximated for each $\omega \in \Omega$, leading to so-called “multicut” algorithms. In addition to optimality cuts, this approach in general requires an additional set of cuts, called *feasibility cuts*, to cut off solutions $x \in X$ for which the recourse problem is infeasible. However, to highlight the differences among solution methods, we here focus on only optimality cuts by assuming *relatively complete recourse*, i.e., the recourse problem is feasible for all $x \in X$ and $\omega \in \Omega$.

When an SIP only involves continuous recourse variables—i.e., all second-stage variables are continuous—the L-shaped method [99] generates optimality cuts using dual solutions of the recourse problem for all scenarios. These cuts are *tight*, meaning that they coincide with the recourse function Q at some points; specifically, when the recourse problem is a linear program and $x = \hat{x}$, the cut derived from the dual solution coincides with Q at \hat{x} . This property ensures that the recourse function can be exactly approximated with finitely many such cuts, resulting in finite convergence of the algorithm.

However, in the presence of integer recourse variables—referred to as the integer recourse—the standard L-shaped method is no longer applicable. To address this, we may convexify the feasible region of the recourse problem and use the LP dual formulation with the convexified feasible region to generate cuts. Carøe and Tind [23] present such an approach, employing *Lift-and-Project cuts* [3] for convexification. Gade et al. [34] propose a similar method, based on *Gomory cuts* [38], and show the finite convergence of their algorithm for two-stage SIPs with binary first-stage variables and integer recourse variables. Zhang et al. [107] further extends this result, handling general integer variables in both stages. Kim and Mehrotra [52] use *mixed-integer rounding cuts* [67] in a similar approach. Notably, Bansal et al [6] present a variant of the L-shaped method for DRO two-stage SIPs, based on a convexification procedure.

Another line of research in Benders decomposition approaches for SIPs focuses on developing

new classes of optimality cuts that do not rely on convexification. For SIPs with binary first-stage variables, Laporte and Louveaux [55] present the integer L-shaped method, which generates optimality cuts using the current objective value and a lower bound on the optimal objective in each iteration; these cuts are referred to as *integer optimality cuts*. This method converges to optimality even with general mixed-integer recourse variables. Zou et al. [111] propose *Lagrangian cuts*, which are derived using a Lagrangian relaxation approach where a copy of variables x is added for each scenario and the linking constraints $x = x(\omega)$ are relaxed. They show that, for an optimal Lagrangian dual solution, the resulting cut is tight. In a different direction, Ahmed et al. [1] develop nonconvex cuts, leveraging Lipschitz continuity and augmented Lagrangian duality, for mixed-integer recourse.

2.1.2 Network Interdiction Problems

We now turn to the literature on NIPs. Despite the extensive body of work on NIPs, our review below focuses primarily on the shortest path interdiction problem (SPIP), which is examined in Chapter 3, as well as a few specific NIPs relevant to the contributions of this dissertation.

Fulkerson and Harding [33] present SPIP, in which the network user's problem is formulated as a shortest path problem. They assume that all variables are continuous; particularly, an interdiction action on an arc can increase the travel cost of the arc by any fractional amount within a given budget. Golden [37] study a variant of SPIP, where the objective is to minimize the total interdicting cost required to ensure a certain amount of increase in the shortest-path travel cost. Israeli and Wood [46] next present SPIP with binary interdiction variables. They propose a Benders decomposition method and *super-valid* inequalities to reduce the solution space of interdiction variables by cutting off non-optimal feasible solutions. A stochastic

model for SPIP is first introduced in [45] and solved by using a Benders decomposition method similar to the forgoing method. Afterward, various variants of stochastic SPIP have been introduced. For example, Hemmecke et al. [42] present a stochastic model, where the objective is to maximize the probability that the shortest-path travel cost exceeds a given threshold. To solve this, they take the dual of the network user's problem, thereby reformulating the whole problem as a single-level maximization problem. In a subsequent paper, Held et al. [40] propose a Benders decomposition method for this SPIP. Nguyen and Smith [68] study stochastic SPIP where random travel costs are uniformly distributed over intervals. They present a *sequential approximation algorithm*, introduced by Cormicn et al. [26], which partitions the intervals into partitions and iteratively refines the number of partitions to tighten the upper and lower bounds on the optimal objective value. In another paper, Nguyen and Smith [69] investigate a risk-averse SPIP with Conditional-Value-at-Risk (CVaR) measure. Song and Shen [95] also study a risk-averse SPIP where the risk preference is modeled by chance constraints, ensuring that the network user's shortest-path cost exceeds a given threshold with high probability. Their model aims to minimize the total interdiction cost while satisfying these chance constraints. Pay et al. [76] present another risk-averse SPIP that uses a utility function to represent the interdictor's risk preference. Its objective is to maximize the expected worst-case utility of the interdictor defined for a known probability distribution. Meanwhile, recent studies have also considered different aspects of deterministic SPIP, such as information asymmetry between players [8, 9, 86], incomplete information [103], and randomized interdiction strategy [44]. We note that none of the aforementioned studies addresses distributional ambiguity in the context of SPIP.

For other types of NIPs, we refer to a survey paper by Smith and Song [95]. Here, we briefly discuss the maximum-reliability path interdiction problem [64, 72] and the maximum flow interdiction problem [101]. The maximum-reliability path interdiction problem has been

studied as an application of SPIP in detecting evaders. In this problem, an interdicator installs sensors on a set of arcs to maximize the probability of detection. A network user (evader) seeks to find a maximum-reliability path from s to t , minimizing the probability of being detected. Pan and Morton [72] propose a Benders decomposition method for the stochastic maximum-reliability path interdiction problem. In the maximum flow interdiction problem, an interdicator seeks to minimize the maximum network flow through interdiction actions that are characterized as removing [101] or diminishing capacities of a set of arcs [102]. The stochastic variants of this problem are presented by Cormican et al. [26] and Janjarassuk and Linderoth [47], where the former proposes a sequential approximation approach, and the latter presents a Benders decomposition method. Lei et al. [56] propose risk-averse maximum flow interdiction problems, in which the objective function is represented using CVaR measures. In their models, the network user can fortify a set of arcs before the realization of uncertainty, as in defender-attacker-defender models in [20]. They incorporate the risk preferences of the network user, either risk-neutral or risk-averse, in these models. Hien et al. [43] study a similar network fortification problem, where the network user aims to protect the network against potential interdiction. They propose a robust stochastic approximation method to solve the problem. Recently, Sadana and Delage [85] explore the maximum flow interdiction problem in DRO, where the objective is to minimize the worst-case of the maximum flow, modeled using CVaR measure. They assume the CVaR measure is associated with the unknown probability distribution of random parameters. They derive a nonconvex reformulation of the problem and then solve it using a spatial branch-and-bound algorithm. Park and Bansal [75] investigate an interdiction problem where the inner problem is a generalized maximum coverage problem. They introduce a decomposition method based on cutting planes derived from properties of the reward objective function. They also present heuristics that achieve significant speedups while yielding effective solutions. In a subsequent paper [74], they extend their study to more generalized problems involving submodular max-

imization in the inner problem. By interpreting the results from both players' perspectives, they provide valuable insights into adversarial learning, particularly in the context of feature selection problems.

2.1.3 Bi-Parameterized Two-Stage Stochastic Programs

The literature on BTSPs is very limited. To the best of our knowledge, existing studies have focused only on continuous settings—where all variables are continuous—and have mainly employed convexification and smoothing approaches for the recourse functions.

Liu et al. [58] propose a decomposition method that applies regularization and smoothing techniques to the recourse function. Specifically, the method involves regularizing the recourse function, deriving its *difference-of-convex* decomposition, and constructing a convex approximation. This approximation serves as a surrogate of the recourse function, allowing the use of a gradient-based method to iteratively update the first-stage solution x . They present a gradient-based decomposition algorithm and show that the algorithm converges to a generalized critical point. Li and Cui [57] also present a decomposition algorithm for two-stage stochastic programs with implicit convex-concave recourse functions where convex approximations of the recourse function are iteratively generated and solved, and the solutions converge to a critical point. Bonze et al. [19] propose a bounding method for a class of bi-parameterized two-stage stochastic nonconvex programs where the objective function involves nonconvex quadratic terms and the feasible region is defined by a simplex. We note that none of the above approaches is directly applicable to BTSPs with integer variables as in our setting.

2.1.4 Multistage Stochastic Integer Programs

In this section, we review the literature on solution methods for multistage stochastic linear programs (MSLPs), multistage stochastic integer programs (MSIPs), and their DRO variants.

Multistage Stochastic Linear Programs.

Nested Benders decomposition (NBD) is a typical decomposition method for solving MSLPs. NBD approximates the cost-to-go function Q_{t+1} at each stage t by a piecewise linear convex function, constructed using optimality cuts. Those cuts are obtained using dual solutions to the subsequent stage's problem. NBD can be implemented more efficiently under the *stage-wise independence* assumption, which states that the random variables at each stage are independent of the random variables at other stages. Specifically, we can employ a scenario sampling approach in each iteration of NBD to mitigate the “curse of dimensionality”. This approach is introduced in [77] and is referred to as stochastic dual dynamic programming (SDDP). It is shown that SDDP finitely converges to optimal decisions for MSLPs with probability one (see [79, 91]). A complexity analysis of SDDP is also provided in [54].

Distributionally Robust MSLPs.

Recently, Philpott et al. [80] consider distributionally robust multistage stochastic linear programs (referred to as DRO-MSLPs) where ambiguity sets are defined using the χ^2 distance from a reference probability distribution. Their approach is based on SDDP and incorporates separation methods to compute worst-case probability distributions for different reference distributions. Duque and Morton [31] present an SDDP-based algorithm for DRO-MSLPs

where ambiguity sets are defined using the Wasserstein metric. Their approaches are based on the strong duality of the inner problem over the Wasserstein ambiguity set and a separation method similar to that of [80]. Park and Bayraksan [73] investigate DRO-MSLPs with ambiguity based on the ϕ -divergence and apply the models to a water allocation problem. They propose an NBD algorithm using the dual reformulation of the inner problem over the ambiguity set.

Multistage Stochastic Integer Programs.

For MSIPs, the standard SDDP is not directly applicable, since LP duality is unavailable. However, these problems can be addressed using a modified SDDP and another decomposition method that leverages Lagrangian duality.

For MSIPs with binary variables, Zou et al. [111] propose an extension of SDDP, referred to as SDDiP. Within SDDiP, the cost-to-go functions are approximated using Lagrangian cuts, which are obtained through a Lagrangian relaxation. They show that the strong duality holds for the Lagrangian dual, thereby ensuring that the resulting Lagrangian cuts are tight to the cost-to-go function. They also provide other cuts and conditions under which SDDiP is guaranteed to converge finitely to optimal decisions.

For a more general class of MSIPs, where all decision variables are allowed to be mixed-integer, there are several studies applying scenario-wise decomposition methods—based on Lagrangian decomposition. For example, Carøe and Schultz [22] present the *dual decomposition* method and propose its implementation in a branch-and-bound procedure. The approach uses a Lagrangian relaxation of *nonanticipativity* constraints, which enforce the scenarios that follow the same history up to stage t to have the same decisions until stage t . Lulli and Sen [59] propose a branch-and-price algorithm, i.e., a branch-and-bound algorithm

with column generation, which can produce lower bounds of the same quality as the dual decomposition method. We also note that, for multistage stochastic mixed-integer nonlinear programs, Zhang and Sun [108] present decomposition methods—one based on NBD and another based on SDDP—which rely on a regularization of the expected cost-to-go function, i.e., $\mathbb{E}_{P_{t+1}}[Q_{t+1}(x_t, \boldsymbol{\xi}_{t+1})]$ at stage t , and a class of cuts called *generalized conjugacy cuts*.

Distributionally Robust MSIPs.

Yu and Shen [106] investigate decision-dependent DRO-MSIPs, where ambiguity sets depend on the decisions made at the previous stage. They consider three types of ambiguity sets constructed based on 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 [65], a dual decomposition approach is presented for a DRO-MSIP where variables can be mixed-integer and ambiguity sets are defined using the Wasserstein metric. The approach takes dual-based reformulations of the inner problems over Wasserstein ambiguity sets in a consecutive manner to derive a single-level minimization equivalent of the DRO-MSIP. Then, they apply a Lagrangian relaxation to the nonanticipativity constraints in the equivalent formulation for deriving a Lagrangian dual. Bayraksan et al. [11] present a decomposition method for DRO-MSIPs, where the scenario tree is divided into subgroups that can be addressed more efficiently. This approach yields bounds on the optimal objective value. They derive conditions for choosing the ambiguity sets' radii for subgroup problems to obtain valid lower bounds.

2.1.5 Distributionally Risk-Receptive Optimization

We now review studies relevant to the DRR framework. In the literature, the DRR framework has been mainly studied in three contexts: (a) making decisions robust to outliers, errors, and adversarial corruptions in a data-driven setting [17, 84], (b) analyzing outcomes based on multiple decision-makers' differing perspectives and varying risk appetites of the decision-makers [49, 74], and (c) obtaining bounds on the true expectations [21, 29].

In the context of robust statistics, Blanchet et al. [17] discuss the connection between the DRR framework and robust statistics, which aims to seek a reliable estimator given samples that may be contaminated. They present the DRR framework as an optimistic optimization approach that first rectifies the contamination and then makes decisions. In [18], as a rectifying-optimizing approach, a DRR program with an ambiguity set defined by an optimal transport distance has been studied. Royset et al. [84] also study a similar optimistic approach based on *Rockafellian relaxations*. Jiang and Xie [48] show that the DRR framework with a specific selection of ambiguity sets can recover many robust statistics, such as the median and the least trimmed squares.

In another direction, DRR programs are utilized in reinforcement learning [94] to find an optimistic policy and Bayesian statistics [70] to approximate a likelihood. In [39], the authors investigate the DRO and DRR frameworks in comparison to a sample average approximation approach for out-of-sample performance. In particular, they show that by solving both DRO and DRR programs, one of their solutions always outperforms the sample average approximation solutions in out-of-sample tests.

Recently, DRR programs are applied to general interdiction problems where the defender's problem involves submodular maximization [74]. In particular, their results in feature selection models show that risk-receptive attackers can devise more effective attacks—reducing

the defender’s test accuracy—compared to risk-neutral or DRO solutions.

In the context of obtaining bounds, Duchi et al. [29] employ the DRR and DRO frameworks to construct a confidence interval for the true optimal objective of a stochastic optimization problem, which can be used to determine the size of the ambiguity sets for a given confidence interval size. Similarly, Cao and Gao [21] address problems involving covariate data where uncertain parameters belong to a specific uncertainty set. They show that solving robust and optimistic optimization problems yields worst-case and best-case rewards, respectively, thus forming a confidence interval for the true reward. Nakao et al. [66] consider a partially observable Markov decision process with distributional ambiguity. They solve a DRR model to obtain an upper bound on the true value function of a DRO partially observable Markov decision process.

To the best of our knowledge, the literature lacks solution approaches for DRR programs involving multistage decision-making with integer variables, despite their applicability to interdiction problems. In addition, we note that decision-dependent uncertainty remains unexplored within the DRR framework, even though it has been extensively studied in the context of stochastic programming and DRO (see, e.g., [7, 41, 60, 106]).

Chapter 3.

Distributionally Risk-Receptive and Robust Network Interdiction Problems with General Ambiguity Set¹

3.1 Introduction

In this chapter, we examine the shortest path interdiction problem (SPIP) under uncertainty and its formulations in the DRR and DRO frameworks. The problem is stated as follows. The interdictor selects a subset of arcs in a network, aiming to disrupt minimum-cost paths of the network user. In the affected network, the network user attempts to find a minimum-cost (i.e., shortest) path from a source node to a destination node. SPIP is one of the most extensively studied NIPs; see, for example, our literature review in Section 2.1.1.

We consider a stochastic SPIP (S-SPIP) that involves uncertainty regarding the success of interdiction attempts and the impact of interdiction on arc's travel cost. S-SPIP can be formulated as a stochastic program with an exact (or estimated) probability distribution. Applying the DRO framework to S-SPIP renders the interdictor's objective to maximize the expected travel cost for the network user under a worst-case distribution within an ambiguity set. We refer to this model as DRO-SPIP. DRO-SPIP arises in applications where the

¹Reference [49]: Sumin Kang and Manish Bansal. Distributionally risk-receptive and robust network interdiction problems with general ambiguity set. *Networks*, 81(1):3–22, 2023.

interdictor is the main protagonist, such as delaying an opponent’s military operations [33, 37] and disrupting illicit supply chains and networks [64, 72, 100]. Conversely, DRR models for SPIP (DRR-SPIPs) are suitable for analyzing the vulnerabilities of critical infrastructure, identifying key network components whose disruption would cause significant damage (see [13, 20, 88] for S-SPIP examples).

DRO-SPIP and DRR-SPIP are generalizations of S-SPIP; when their ambiguity sets are singletons, they reduce to S-SPIP. Therefore, the optimal objective values of DRO-SPIP and DRR-SPIP provide lower and upper bounds, respectively, of the objective value obtained from solving S-SPIP. In addition to generalizing S-SPIP, DRO-SPIP and DRR-SPIP allow adjustments on the level of robustness (or risk-aversion) and risk-receptiveness, respectively, of the interdictor. These adjustments are managed by choice of ambiguity set size, which can range from a singleton set to the set of all probability distributions supported on Ω . When the largest ambiguity set (denoted by $\hat{\mathcal{P}}$) is used, DRO-SPIP yields the most conservative solution for the interdictor. By selecting subsets of $\hat{\mathcal{P}}$, we can obtain less conservative solutions reflecting the decreased level of risk-aversion of the interdictor. Conversely, DRR-SPIP with $\hat{\mathcal{P}}$ as the ambiguity set provides the most optimistic solution for the interdictor (or the most conservative solution for the network user), since it places greater weights on the scenarios most unfavorable to the network user. We can also decrease the level of risk-receptiveness of the interdictor by using subsets of $\hat{\mathcal{P}}$.

3.2 Problem Definition: DRO-SPIP and DRR-SPIP

We now formally describe the problem of interest as follows. Consider a directed network $G = (N, A)$. A source node and a destination node along the network user’s path are denoted by s and t , respectively. For each arc $a = (i, j) \in A$ between nodes i and $j \in N$, denote its travel

cost by c_a . Also, the penalty cost vector and the success of interdiction vector are denoted by random vectors $\mathbf{d} \in \mathbb{R}_+^{|A|}$ and $\boldsymbol{\zeta} \in \{0, 1\}^{|A|}$, respectively. We assume that $(\mathbf{d}, \boldsymbol{\zeta})$ has finite support, and its realization (d^ω, ζ^ω) , for each scenario $\omega \in \Omega$, occurs with probability p_ω . The interdictor's decision is denoted by a binary decision vector $x \in \{0, 1\}^{|A|}$ where $x_a = 1$ or $x_a = 0$ implies that arc $a \in A$ is interdicted or not interdicted, respectively. A set of all feasible interdiction decision vectors is represented as $X := \{x \in \{0, 1\}^{|A|} \mid \sum_{a \in A} x_a = b\}$ where $b > 0$ denotes the interdicting budget. Note that the network user is a “wait-and-see” player who makes a decision after observing the realizations of uncertainty and the interdictor's decision. Thus, the network user's shortest path problem in each scenario $\omega \in \Omega$ involves the deterministic travel costs for all arcs which are represented as $(c_a + d_a^\omega \zeta_a^\omega \hat{x}_a)$ for $a \in A$ where \hat{x}_a is a given interdiction decision. Note that the penalty term $(d_a^\omega \zeta_a^\omega \hat{x}_a)$ is activated when both \hat{x}_a and ζ_a^ω have values of 1, which means that the interdictor decides to interdict arc $a \in A$, and the interdiction is successful in scenario $\omega \in \Omega$. The network user's decision variable $y_a^\omega \in \{0, 1\}$ represents whether arc $a \in A$ is traversed in the shortest path in scenario $\omega \in \Omega$ or not. For each node $i \in N$, let $\delta^+(i)$ and $\delta^-(i)$ be the set of outgoing and incoming arcs for node i , respectively. We then formulate a risk-neutral stochastic (shortest or minimum-cost path) network interdiction problem as follows:

$$(S-SPIP) \quad \max_{x \in X} \sum_{\omega \in \Omega} p_\omega \left(\min_{y^\omega} \sum_{a \in A} (c_a + d_a^\omega \zeta_a^\omega x_a) y_a^\omega \right), \quad (3.1a)$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(i)} y_a^\omega - \sum_{a \in \delta^-(i)} y_a^\omega = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{otherwise} \end{cases}, \quad \forall i \in N, \omega \in \Omega, \quad (3.1b)$$

$$y_a^\omega \in [0, 1], \quad \forall a \in A, \omega \in \Omega. \quad (3.1c)$$

Notice that since the constraint matrix associated with constraints (3.1b), i.e., flow balance constraints, of the inner shortest path problem is totally unimodular, we can let $y_a \in [0, 1]$ instead of being binary [12]. Also, we denote the set of feasible solutions of the network user by $Y^\omega := \{y^\omega \mid (3.1b) \text{ and } (3.1c) \text{ hold}\}$ for $\omega \in \Omega$.

We now present formulations of DRO-SPIP and DRR-SPIP as follows:

$$\text{(DRO-SPIP)} \quad \max_{x \in X} \min_{P \in \mathcal{P}} \sum_{\omega \in \Omega} p_\omega \min_{y^\omega \in Y^\omega} \sum_{a \in A} (c_a + d_a^\omega \zeta_a^\omega x_a) y_a^\omega; \quad (3.2)$$

$$\text{(DRR-SPIP)} \quad \max_{x \in X} \max_{P \in \mathcal{P}} \sum_{\omega \in \Omega} p_\omega \min_{y^\omega \in Y^\omega} \sum_{a \in A} (c_a + d_a^\omega \zeta_a^\omega x_a) y_a^\omega, \quad (3.3)$$

where P denotes the probability distribution, $\{p_\omega\}_{\omega \in \Omega}$. In DRO-SPIP (3.2), the objective is to maximize the expected travel cost under the worst-case probability distribution within \mathcal{P} that is identified by solving the minimization over \mathcal{P} . On the contrary, in DRR-SPIP (3.3), our focus is on the best-case probability distribution within \mathcal{P} that is obtained by maximizing over \mathcal{P} . Notice that for $|\mathcal{P}| = 1$, both of these models reduce to the (risk-neutral) S-SPIP, and for $|\Omega| = 1$, they further reduce to the deterministic NIP. Likewise, when \mathcal{P} is defined by all probability distributions supported on Ω , DRO-SPIP and DRR-SPIP reduce to robust optimization models where interdictor makes robust decisions for the worst-case scenario and risk-receptive decisions for the best-case scenario, respectively.

We note that our approaches can accommodate random travel costs c and random network topology (e.g., random location of source node s and destination node t), although these are not explicitly considered in this work.

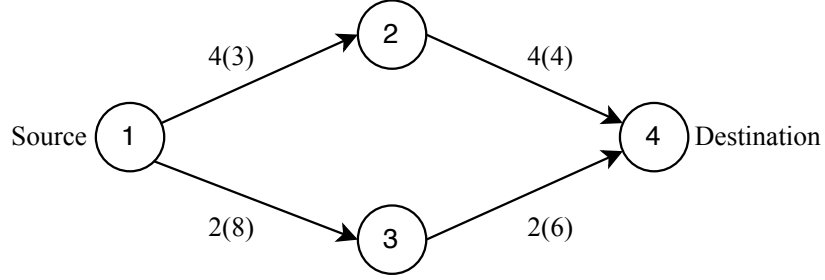


Figure 3.1: Example network with source node $s = 1$ and destination node $t = 4$.

3.2.1 Illustrative Example

To illustrate the problems, we consider a toy example with a network $G = (N, A)$ in Figure 3.1 where $N = \{1, 2, 3, 4\}$, $A = \{(1, 2), (1, 3), (2, 4), (3, 4)\}$, and nodes 1 and 4 are source s and destination t nodes, respectively. The travel costs $c_{12} = 4$, $c_{13} = 2$, $c_{24} = 4$, and $c_{34} = 2$ are known. In this example, we assume the uncertain success of interdiction attempts and consider two realizations: $\zeta^{\omega_1} = (1, 1, 0, 0)$ and $\zeta^{\omega_2} = (0, 0, 1, 1)$ corresponding to the set A . For the sake of convenience, in this example, we assume penalty costs $d_{12}^\omega = 3$, $d_{13}^\omega = 8$, $d_{24}^\omega = 4$, and $d_{34}^\omega = 6$ for all $\omega \in \{\omega_1, \omega_2\}$, and consider a finite ambiguity set $\mathcal{P} = \{P_1, P_2, P_3\} = \{(0.7, 0.3), (0.3, 0.7), (0.5, 0.5)\}$. Notice that the shortest (or minimum-cost) path between s and t on this network when there is no interdiction, is a path traversing along arcs $(1, 3)$ and $(3, 4)$, which we denote by SP^* , that costs 4 units.

Let the interdiction budget b be equal to 2 units. This implies that the interdictor has to select two arcs in A , and therefore, she has six possible choices: $\{(1, 2), (1, 3)\}$, $\{(2, 4), (3, 4)\}$, $\{(1, 3), (3, 4)\}$, $\{(1, 2), (2, 4)\}$, $\{(1, 2), (3, 4)\}$, and $\{(1, 3), (2, 4)\}$, i.e., the set $X = \{x^i\}_{i=1}^6 = \{(1, 1, 0, 0), (0, 0, 1, 1), (0, 1, 0, 1), (1, 0, 1, 0), (1, 0, 0, 1), (0, 1, 1, 0)\}$. For a given $x \in X$ and $P \in \mathcal{P}$, we solve the shortest path problem for each scenario and report the expected shortest path cost in rows 2-4 of Table 3.1. Using these expected costs, we also find an optimal interdiction solution corresponding to the following risk preference of the interdictor.

Table 3.1: Expected travel cost for a given interdicator's action $x \in X = \{x^1, \dots, x^6\}$ and probability distribution $P \in \{P_1, P_2, P_3\}$; optimal solution and objective value of S-SPIP for each distribution P , DRR-SPIP, and DRO-SPIP.

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

- Robust (DRO-SPIP):** Consider a risk-averse interdicator who chooses to interdict arcs $(1, 3)$ and $(3, 4)$ in an optimal solution x^3 . Note that both of these arcs are on path SP^* . The interdiction will successfully affect the cost to travel arc $(1, 3)$ in scenario ω_1 and arc $(3, 4)$ in scenario ω_2 . As a result, the total travel cost of SP^* increases by 8 and 6 units for scenarios ω_1 and ω_2 , respectively. Accordingly, an alternative path along arcs $(1, 2)$ and $(2, 4)$ that costs 8 units becomes a new shortest path in both scenarios. From the interdicator's perspective, this choice reflects a risk-averse strategy, as it ensures the interdiction of the original shortest path SP^* , thereby preventing the network user from using this path in both scenarios.
- Risk-Receptive (DRR-SPIP):** If we assume that the interdicator is risk-receptive, then the optimal strategy is to interdict arcs $(1, 2)$ and $(1, 3)$, i.e., x^1 . Both the arcs will be successfully interdicted in scenario ω_1 , thereby resulting in a shortest-path cost of 11 units. In contrast, the interdiction attempts will fail in scenario ω_2 , and thus SP^* remains the shortest path with a cost of 4 units. Observe that this interdiction solution is worse for the worst-case probability distribution in comparison to the DRO-SPIP solution. However, from the network user's perspective, this result shows the most pessimistic scenario when the interdicator is most successful, which is not observable in

the robust model.

- **Risk-Neutral (S-SPIP):** Observe that the optimal interdiction solution for S-SPIP changes as we vary the known probability distribution, i.e., x^1 (8.9 units) is optimal for P_1 , x^2 (8.2 units) is optimal for P_2 , and x^3 (8 units) is optimal for P_3 distribution. The optimal solution values range from DRO-SPIP's optimal solution value to the DRR-SPIP's optimal solution value.

3.3 Solution Approaches for DRR-SPIP

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

$$z_{RR}^{opt} := \max_{x \in X} D_{max}(x), \quad (3.4a)$$

$$D_{max}(x) := \max_{P \in \mathcal{P}} \sum_{\omega \in \Omega} p_{\omega} Q_{\omega}(x), \quad (3.4b)$$

$$Q_{\omega}(x) := \min_{y^{\omega} \in Y^{\omega}} \sum_{a \in A} (c_a + d_a^{\omega} \zeta_a^{\omega} x_a) y_a^{\omega}. \quad (3.4c)$$

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

3.3.1 Reformulation-Based Approach for DRR-SPIP

Combining the outer and inner maximization, formulation (3.4) is equivalent to

$$\max \sum_{\omega \in \Omega} p_\omega Q_\omega(x), \quad (3.5a)$$

$$\text{s.t. } x \in X, \{p_\omega\}_{\omega \in \Omega} \in \mathcal{P}, \quad (3.5b)$$

where we treat $\{p_\omega\}_{\omega \in \Omega}$ as first-stage variables. This leads to bilinear terms $p_\omega x_a$ in (3.5a) since $p_\omega Q_\omega(x) = \min_{y^\omega \in Y^\omega} \sum_{a \in A} (p_\omega c_a + d_a^\omega \zeta_a^\omega p_\omega x_a) y_a^\omega$, for $\omega \in \Omega$. We linearize these bilinear terms as follows. Since the interdiction variables x_a , $a \in A$, are binary and $p_\omega \in [0, 1]$, $\omega \in \Omega$, we introduce a nonnegative continuous variable $\eta_a^\omega \in \mathbb{R}_+$ that satisfies $\eta_a^\omega \leq p_\omega$ and $\eta_a^\omega \leq x_a$ for $\omega \in \Omega$, $a \in A$, so that $p_\omega x_a$ can be replaced by η_a^ω . This results in the following reformulation of DRR-SPIP:

$$\max \sum_{\omega \in \Omega} R_\omega(p_\omega, \eta^\omega), \quad (3.6a)$$

$$\text{s.t. } \eta_a^\omega \leq x_a, \quad \forall \omega \in \Omega, a \in A, \quad (3.6b)$$

$$\eta_a^\omega \leq p_\omega, \quad \forall \omega \in \Omega, a \in A, \quad (3.6c)$$

$$\eta_a^\omega \geq 0, \quad \forall \omega \in \Omega, a \in A, \quad (3.6d)$$

$$x \in X, \{p_\omega\}_{\omega \in \Omega} \in \mathcal{P}, \quad (3.6e)$$

where $\eta^\omega := \{\eta_a^\omega\}_{a \in A}$ and $R_\omega(p_\omega, \eta^\omega) := \min \left\{ \sum_{a \in A} (p_\omega c_a + d_a^\omega \zeta_a^\omega \eta_a^\omega) y_a^\omega : y^\omega \in Y^\omega \right\}$ for $\omega \in \Omega$. Note that, for a given y^ω , $R_\omega(p_\omega, \eta^\omega)$ is an affine function of p_ω and η^ω . This indicates that $R_\omega(p_\omega, \eta^\omega)$ is a point-wise minimum of affine functions and is thus a piecewise linear concave function. Formulation (3.6) can be solved using an L-shaped method with branch-and-cut framework [55, 99] for two-stage stochastic programs with continuous recourse, i.e., $R_\omega(\cdot)$ in

our case, and mixed-binary first-stage variables $(x, \{p_\omega, \eta^\omega\}_\omega)$. In this approach, so-called optimality cuts for $R_\omega(p_\omega, \eta^\omega)$, i.e.,

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

are derived using the extreme points \hat{y}^ω of Y^ω to get outer approximations of R_ω for $\omega \in \Omega$. Note that this method is applicable for a general ambiguity set \mathcal{P} . For an ambiguity set \mathcal{P} defined as a polyhedral set [4, 5, 6] (for example, moment matching set [27], Wasserstein ambiguity set [78], or total variation distance set [82]) or a mixed-binary set, the feasible region of (3.6) is a mixed-binary set. The reader can refer to Appendix A.2 for more details on the implementation of this approach (used for our computational experiments).

3.3.2 Cuts-Based Decomposition Approach for DRR-SPIP

We present another decomposition approach for DRR-SPIP where we utilize a distribution separation procedure and a class of valid inequalities to obtain outer approximations of the nonconcave objective function $D_{max}(x)$ within a Benders' decomposition framework.

Theorem 3.1. *For a given interdiction solution $\hat{x} = \{\hat{x}_a\}_{\omega \in \Omega}$ and the network user's optimal solutions $\hat{y}^\omega = \{\hat{y}_a^\omega\}_{a \in A}$ for $Q_\omega(\hat{x})$, for $\omega \in \Omega$, the following inequality is valid for all $x \in X$.*

$$D_{max}(x) \leq D_{max}(\hat{x}) + \sum_{a \in A} \beta_a(\hat{x}_a)(x_a - \hat{x}_a), \quad (3.7)$$

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

$$\beta_a(\hat{x}_a) = \begin{cases} \max_{P \in \mathcal{P}} \sum_{\omega \in \Omega} p_\omega d_a^\omega \zeta_a^\omega \hat{y}_a^\omega, & \text{if } \hat{x}_a = 0 \\ \min_{P \in \mathcal{P}} \sum_{\omega \in \Omega} p_\omega d_a^\omega \zeta_a^\omega \hat{y}_a^\omega, & \text{if } \hat{x}_a = 1 \end{cases}, \quad \forall a \in A. \quad (3.8)$$

Remark 3.1. For a deterministic NIP, i.e., when $|\Omega| = 1$, coefficient $\beta_a(\hat{x}_a) = d_a^\omega \zeta_a^\omega \hat{y}_a^\omega$, $a \in A$. In this special case, the proposed cut reduces to the so-called *global Benders cut*, which was introduced for a deterministic electric power grid interdiction problem [87].

Inequalities (3.7) defined for a set of interdiction solutions provide a polyhedral outer approximation of the hypograph of the function $D_{max}(x)$. We now present a decomposition algorithm to solve DRR-SPIP to optimality, in which we iteratively construct tighter outer approximations of $D_{max}(x)$. Algorithm 3.1 provides a pseudocode for this approach and it works as follows. To initialize the algorithm, we set iteration counter L to 1, lower bound $z_{RR}^{lb} = -\infty$, and upper bound $z_{RR}^{ub} = \infty$. We also select an initial feasible solution $\hat{x}^1 \in X$. At each iteration $L \geq 1$, we solve a master problem (denoted by \mathcal{M}_{RR}^L), which is defined as

$$z_{RR}^{ub} := \max_{x \in X, \theta} \theta \quad (3.9a)$$

$$\text{s.t. } \theta \leq D_{max}(\hat{x}^l) + \sum_{a \in A} \beta_a(\hat{x}_a^l)(x_a - \hat{x}_a^l), \quad \text{for } l = 1, \dots, L, \quad (3.9b)$$

where $\hat{x}^l \in X$ is an optimal solution to the master problem \mathcal{M}_{RR}^{l-1} for $l \in \{2, \dots, L\}$. Constraints (3.9b) are referred to as optimality cuts for (hypograph of) $D_{max}(x)$. To derive an optimality cut (3.9b) and a lower bound, we solve the network user's shortest path problem (3.4c), referred to as subproblem, for the given \hat{x}^L to obtain an optimal solution $\hat{y}^{\omega, L}$ and compute $Q_\omega(\hat{x}^L)$, for $\omega \in \Omega$ (Line 3). In Line 4, we solve a distribution separation problem, which is defined as

$$z_{RR}^{dsp, L} := \max_{P \in \mathcal{P}} \sum_{\omega \in \Omega} p_\omega Q_\omega(\hat{x}^L), \quad (3.10)$$

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

Algorithm 3.1: A decomposition algorithm for DRR-SPIP

```

1 Initialization:  $L \leftarrow 1$ ;  $z_{RR}^{lb} \leftarrow -\infty$ ;  $z_{RR}^{ub} \leftarrow \infty$ ;  $\hat{x}^1 \in X$  (an initial feasible solution);
2 while  $(z_{RR}^{ub} - z_{RR}^{lb})/z_{RR}^{lb} > \epsilon$  do
3   Solve subproblem (3.4c) to obtain optimal solution  $\hat{y}^{\omega,L}$  and  $Q_\omega(\hat{x}^L) \forall \omega \in \Omega$ ;
4   Solve distribution separation problem (3.10) to obtain  $\{\hat{p}_\omega^L\}_{\omega \in \Omega}$  and
       $z_{RR}^{dsp,L} = D_{max}(\hat{x}^L)$ ;
5   if  $(z_{RR}^{dsp,L} > z_{RR}^{lb})$  then
6      $(z_{RR}^{lb}, \hat{x}^*, \{\hat{y}^{\omega,*}\}_{\omega \in \Omega}) \leftarrow (z_{RR}^{dsp,L}, \hat{x}^L, \{\hat{y}^{\omega,L}\}_{\omega \in \Omega})$ ;
      /* Update best-known lower bound and feasible solution */
7   end
8   Compute  $\beta_a(\hat{x}_a^L)$ ,  $a \in A$ , using (3.8) and add an optimality cut, i.e., (3.7) with
       $\hat{x} = \hat{x}^L$ , to  $\mathcal{M}_{RR}^{L-1}$  to get  $\mathcal{M}_{RR}^L$ ;
9   Solve master problem  $\mathcal{M}_{RR}^L$  and obtain its optimal solution  $(\hat{x}^{L+1}, \hat{\theta}^{L+1})$ ;
10  Update upper bound  $z_{RR}^{ub} \leftarrow \hat{\theta}^{L+1}$ ;
11   $L \leftarrow L + 1$ ;
12 end
13 Return:  $\hat{x}^*$ ,  $\{\hat{y}^{\omega,*}\}_{\omega \in \Omega}$  and  $z_{RR}^{lb}$ .

```

Then, in Line 8, we generate an optimality cut (3.7) for \hat{x}^L as proposed in Theorem 3.1. Using $\hat{y}^{\omega,L}$, cut coefficients $\beta_a(\hat{x}_a)$, $a \in A$, are computed by solving (3.8) corresponding to $\hat{x} = \hat{x}^L$ using the distribution separation algorithm. This generated cut is added to \mathcal{M}_{RR}^{L-1} to get an updated master problem \mathcal{M}_{RR}^L (a tighter outer approximation) for $L \geq 1$ where \mathcal{M}_{RR}^0 is defined by (3.9a). We solve the updated master problem (a mixed-binary linear program), store optimal solution $(\hat{x}^{L+1}, \hat{\theta}^{L+1})$, and update the best-known upper bound z_{RR}^{ub} to $\hat{\theta}^{L+1}$ (Lines 9 and 10). After that, if the optimality gap between bounds $(z_{RR}^{ub} - z_{RR}^{lb})/z_{RR}^{lb}$ is not greater than a given tolerance level $\epsilon > 0$, we terminate the algorithm and return the best-known feasible solution and lower bound (Lines 2 and 13). Otherwise, we proceed to iteration $L + 1$.

Theorem 3.2. *Algorithm 3.1 solves DRR-SPIP to optimality in a finite number of iterations if there exists a finitely convergent oracle for the distribution separation problem (3.10).*

Remark 3.2. There exists a finitely convergent oracle for the distribution separation problem

associated with the ambiguity set defined as either a polyhedral set or a mixed-binary set. For a mixed-binary ambiguity set, the lift-and-project algorithm proposed by Balas et al. [3] can be used to solve the distribution separation problem.

Remark 3.3. According to Algorithm 3.1, a mixed-binary linear program \mathcal{M}_{RR}^L has to be solved in each iteration $L \geq 1$. In our implementation for computational experiments, we have embedded this procedure in a branch-and-cut framework, which works as follows. We construct a single branch-and-cut tree. At each node of this tree, we solve a linear programming relaxation (node-LP) of the master problem (3.9). If the values of the interdiction solution to the node-LP are integral, Steps 3-4, and 8 of Algorithm 3.1 are performed, and an optimality cut is added to the master problem if it is violated. Otherwise, we continue the typical branch-and-cut procedure, i.e., branching based on fractional values of the first-stage variables, node pruning based on bounds, and depth-first search traversal.

3.3.3 Restrict-and-Append Algorithm

We propose a “restrict-and-append” algorithm for DRR-SPIP where we iteratively solve a restriction of DRR-SPIP defined for a finite subset of \mathcal{P} and then append the subset of distributions, if needed, by adding a new distribution. In iteration $L \geq 1$, we use Algorithm 1 in a branch-and-cut framework to solve DRR-SPIP with set $\bar{\mathcal{P}}^L$ as ambiguity set. Due to the finiteness of set $\bar{\mathcal{P}}^L$ and no closed-form representation, we solve the distribution separation problem by explicitly enumerating all distributions in $\bar{\mathcal{P}}^L$. Using $\bar{\mathcal{P}}^L$ as ambiguity set restricts the feasible region of the inner maximization in (3.4b), thereby leading to a restricted DRR-SPIP,

$$z_{RR}^{lb} = \max_{x \in X, P \in \bar{\mathcal{P}}^L} \sum_{\omega \in \Omega} p_{\omega} Q_{\omega}(x). \quad (3.11)$$

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

Remark 3.4. Note that the restrict-and-append algorithm is a heuristic, and it is not guaranteed to produce an optimal solution.

3.4 Exact Solution Approaches for DRO-SPIP

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

$$z_{RO}^{opt} := \max_{x \in X} D_{min}(x), \quad (3.12a)$$

$$D_{min}(x) := \min_{P \in \mathcal{P}} \sum_{\omega \in \Omega} p_{\omega} Q_{\omega}(x), \quad (3.12b)$$

$$Q_{\omega}(x) := \min_{y^{\omega} \in Y^{\omega}} \sum_{a \in A} (c_a + d_a^{\omega} \zeta_a^{\omega} x_a) y_a^{\omega}. \quad (3.12c)$$

Here, the value function $Q_{\omega}(x)$ provides the shortest-path travel cost for a given $x \in X$ and $\omega \in \Omega$. Notice that function $Q_{\omega}(x)$ is concave on the convex hull of X . Therefore, for a given probability distribution $\{p_{\omega}\}_{\omega \in \Omega} \in \mathcal{P}$, the convex combination of the concave functions

$Q_\omega(x)$ for $\omega \in \Omega$, i.e., $\sum_{\omega \in \Omega} p_\omega Q_\omega(x)$, is also concave. This concludes that the value function $D_{min}(x)$, is also concave because it is point-wise minimum of concave functions. Furthermore, problem (3.12) is also equivalent to solving the following semi-infinite optimization problem:

$$\max_{x \in X, \theta} \theta \tag{3.13a}$$

$$\text{s.t. } \theta \leq \sum_{\omega \in \Omega} p_\omega \sum_{a \in A} (c_a + d_a^\omega \zeta_a^\omega x_a) \hat{y}_a^\omega, \quad \forall \hat{y}^\omega \in \hat{Y}_E^\omega, \omega \in \Omega, \text{ and } \{p_\omega\}_{\omega \in \Omega} \in \mathcal{P}, \tag{3.13b}$$

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

3.4.1 Decomposition Approach for DRO-SPIP

We present a decomposition algorithm for DRO-SPIP in which we iteratively obtain outer approximations of the function $D_{min}(x)$. A pseudocode of the proposed decomposition procedure is given in Algorithm 3.2, which works as follows. We start the algorithm using an initial feasible solution $\hat{x}^1 \in X$. We also initialize iteration counter L , lower bound z_{RO}^{lb} , and upper bound z_{RO}^{ub} to 1, $-\infty$, and ∞ , respectively. For each iteration $L \geq 1$, we compute $Q_\omega(\hat{x}^L)$ by solving subproblem (3.12c) for all $\omega \in \Omega$ and store their optimal solutions $\hat{y}^{\omega,L}, \omega \in \Omega$ (Line 3). Using $Q_\omega(\hat{x}^L)$, we then solve a distribution separation problem $D_{min}(\hat{x}^L)$, i.e.,

$$z_{RO}^{dsp,L} := \min_{P \in \mathcal{P}} \sum_{\omega \in \Omega} p_\omega Q_\omega(\hat{x}^L), \tag{3.14}$$

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

Algorithm 3.2: A decomposition algorithm for DRO-SPIP

```

1 Initialization:  $L \leftarrow 1$ ;  $z_{RO}^{lb} \leftarrow -\infty$ ;  $z_{RO}^{ub} \leftarrow \infty$ ,  $\hat{x}^1 \leftarrow \text{any } x \in X$ ;
2 while  $(z_{RO}^{ub} - z_{RO}^{lb})/z_{RO}^{lb} > \epsilon$  do
3   Solve subproblem (3.12c) to obtain optimal solution  $\hat{y}^{\omega,L}$  and  $Q_\omega(\hat{x}^L) \forall \omega \in \Omega$ ;
4   Solve distribution separation problem (3.14) to obtain  $\{\hat{p}_\omega^L\}_{\omega \in \Omega}$  and
       $z_{RO}^{dsp,L} = D_{min}(\hat{x}^L)$ ;
5   if  $(z_{RO}^{dsp,L} > z_{RO}^{lb})$  then
6      $(z_{RO}^{lb}, \hat{x}^*, \{\hat{y}^{\omega,*}\}_{\omega \in \Omega}) \leftarrow (z_{RO}^{dsp,L}, \hat{x}^L, \{\hat{y}^{\omega,L}\}_{\omega \in \Omega})$ ;
      /* Update best-known lower bound and feasible solution */
7   end
8   Add an optimality cut to  $\mathcal{M}_{RO}^{L-1}$  to get  $\mathcal{M}_{RO}^L$ ;
9   Solve master problem  $\mathcal{M}_{RO}^L$  and obtain its optimal solution  $(\hat{x}^{L+1}, \hat{\theta}^{L+1})$ ;
10  Update upper bound  $z_{RO}^{ub} \leftarrow \hat{\theta}^{L+1}$ ;
11   $L \leftarrow L + 1$ ;
12 end
13 Return:  $\hat{x}^*$ ,  $\{\hat{y}^{\omega,*}\}_{\omega \in \Omega}$  and  $z_{RO}^{lb}$ .

```

and $\{\hat{y}^{\omega,L}\}_{\omega \in \Omega}$, i.e.,

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

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

$$z_{RO}^{ub} := \max_{x \in X, \theta} \theta \tag{3.15a}$$

$$\text{s.t. } \theta \leq \sum_{\omega \in \Omega} \hat{p}_\omega^l \sum_{a \in A} (c_a + d_a^\omega \zeta_a^\omega x_a) \hat{y}_a^{\omega,l}, \quad \forall l = 1, \dots, L-1, \tag{3.15b}$$

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

Remark 3.5. In Algorithm 3.2, \mathcal{M}_{RO}^L is a mixed-binary program and solvable using any off-

the-shelf solver for mixed-integer programs. However, we have embedded the decomposition procedure in a branch-and-cut framework for our computational experiments, which works as follows. At each node of the branch-and-cut tree, we solve a linear programming relaxation (node-LP) of the relaxed master problem (3.15). If the values of the interdiction solution to the node-LP are integral, Steps 3-4 of Algorithm 3.2 are performed, and an optimality cut is added to the master problem if it is violated. Otherwise, we continue the typical branch-and-cut procedure.

Theorem 3.3. *Algorithm 3.2 is finitely convergent to an optimal solution of DRO-SPIP if there exists a finitely convergent oracle for the distribution separation problem (3.14).*

3.4.2 Relax-and-Append Algorithm for DRO-SPIP

We present a “relax-and-append” algorithm to solve DRO-SPIP which is similar to the restrict-and-append algorithm for DRR-SPIP. More specifically in iteration $L \geq 1$, we use a branch-and-cut algorithm with Algorithm 3.2 to solve DRO-SPIP with a finite set $\bar{\mathcal{P}}^L \subset \mathcal{P}$ as ambiguity set. Since $\bar{\mathcal{P}}^L$ is a finite set without any closed-form representation, we solve the distribution separation problem by explicitly enumerating all distributions. Note that solving DRO-SPIP with $\bar{\mathcal{P}}^L$ will produce a relaxed solution for the original DRO-SPIP which yields an upper bound on z_{RO}^{opt} :

$$z_{RO}^{ub,L} = \max_{x \in X} \min_{P \in \bar{\mathcal{P}}^L} \sum_{\omega \in \Omega} p_{\omega} Q_{\omega}(x). \quad (3.16)$$

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

tribution obtained by solving the outer distribution separation problem to $\bar{\mathcal{P}}^L$ and proceed to iteration $L + 1$.

Theorem 3.4. *The relax-and-append algorithm solves DRO-SPIP to optimality in finitely many iterations, if there exists a finitely convergent oracle to solve the outer distribution separation problem (3.14).*

Remark 3.6. In contrast to the restrict-and-append algorithm, the relax-and-append algorithm is guaranteed to produce an optimal solution.

3.5 Computational Results

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

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

$q = 1, \dots, m$. Then, the moment matching set is defined as

$$\mathcal{P}_M := \left\{ P = \{p_\omega\}_{\omega \in \Omega} : \underline{u}_q \leq \sum_{\omega \in \Omega} p_\omega f_q(\omega) \leq \bar{u}_q, \text{ for } q = 1, \dots, m; \right. \\ \left. \sum_{\omega \in \Omega} p_\omega = 1; p_\omega \geq 0, \forall \omega \in \Omega \right\}, \quad (3.17)$$

where \underline{u}_q and \bar{u}_q denote the predetermined component-wise lower and upper bounds, respectively. For experiments, we consider component-wise moment inequalities for ζ , i.e.,

$$f_1(\omega) = \zeta_1^\omega, f_2(\omega) = \zeta_2^\omega, \dots, f_{|A|}(\omega) = \zeta_{|A|}^\omega, \\ f_{|A|+1}(\omega) = (\zeta_1^\omega)^2, f_{|A|+2}(\omega) = (\zeta_2^\omega)^2, \dots, f_{2 \times |A|}(\omega) = (\zeta_{|A|}^\omega)^2.$$

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

$$\mathcal{P}_W := \left\{ P = \{p_\omega\}_{\omega \in \Omega} : \sum_{\omega \in \Omega} p_\omega = 1; \sum_{\omega_i \neq \omega_j \in \Omega} \|f(\omega_i) - f(\omega_j)\|_1 v_{\omega_i, \omega_j} \leq \epsilon_W; \right. \\ \sum_{\omega_j \in \Omega} v_{\omega_i, \omega_j} = p_{\omega_i}, \forall \omega_i \in \Omega; \sum_{\omega_i \in \Omega} v_{\omega_i, \omega_j} = p_{\omega_j}^*, \forall \omega_j \in \Omega; \quad (3.18) \\ \left. p_\omega \geq 0, \forall \omega \in \Omega; v_{\omega_i, \omega_j} \geq 0, \forall \omega_i, \omega_j \in \Omega \right\}.$$

Since \mathcal{P}_M and \mathcal{P}_W are polyhedral sets, the corresponding distribution separation problems

are linear programs, and to solve them, we use Gurobi's primary simplex algorithm because it performed better than the others in our preliminary tests for DRR-SPIP and DRO-SPIP instances.

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

3.5.1 Instance Generation and Experiment Setup

We generate DRR-SPIP and DRO-SPIP test instances based on the instances from Nguyen and Smith [68], with some modifications. We consider networks with 40, 60, 80, and 100 nodes, and label them N40, N60, N80, and N100, respectively. In the instances of [68], around 20% of all arcs in the network have the travel cost uncertainty where $c_a^\omega \in [c_a^L, c_a^U]$ for $a \in A$. To modify these instances for DRR-SPIP and DRO-SPIP, we set $c_a = 0.5(c_a^L + c_a^U)$ and randomly generate interdiction successes $\zeta_a^\omega, \omega \in \Omega$ (for arcs with uncertain cost in the original instances) from a Bernoulli distribution with the probability of success of 0.75. For the remaining 80% of the arcs, $\zeta_a^\omega = 1$ for all $\omega \in \Omega$. The impact of successful interdiction on each arc, $d_a, a \in A$, is deterministic and directly taken from the instances of [68]. We use label US20 for these instances. In addition, we generate instances where $\zeta_a^\omega, \omega \in \Omega$ is uncertain for all $a \in A$ and label them as US100. For each instance category, Nxx and USxx, we perform tests with different interdiction budgets $b \in \{2, 5, 10\}$, scenario sizes $|\Omega| \in \{100, 500, 1000\}$, and ambiguity sets $\mathcal{P} \in \{\mathcal{P}_M, \mathcal{P}_W\}$. For \mathcal{P}_M , we compute the first moment of random variable, i.e., $u_a = \sum_{\omega \in \Omega} \zeta_a^\omega / |\Omega|, a \in A$. Then, given tolerance level $\epsilon_M = 0.05$, we compute the bounds, $\underline{u}_a = (1 - \epsilon_M)u_a$ and $\bar{u}_a = (1 + \epsilon_M)u_a, a \in A$. Likewise, for \mathcal{P}_W , we set ϵ_W equal to ρ ($= 0.1$, by default) times the average of the L_1 -norm distance

between all observations ω_i and ω_j in Ω .

3.5.2 Computational Results for DRR-SPIP

In Tables 3.2 and 3.3, we report the computational results using the proposed algorithms for US20 and US100 instances, respectively, of DRR-SPIP with the moment matching set and the Wasserstein ambiguity set. The columns corresponding to results for the reformulation-based algorithm, the cut-based decomposition algorithm, and the restrict-and-append algorithm are labeled as “Reformulation,” “Decomposition,” and “RestrictAppend,” respectively. Each row reports average results for ten instances. For each algorithm, we report the average computational time, T in seconds, for the instances solved with the time limit of 1 hour (3600 seconds). For the instances unsolved within the time limit, we report the average optimality gap (Gap %) and the number of unsolved instances out of ten in parentheses. When all ten instances are solved within the time limit, we use label “ZG” to denote zero optimality gap. In contrast, if none of the ten instances is solved within the time limit, then we use label “3600+.” Since the restrict-and-append algorithm is a heuristic, we present a relative gap between the best solution value obtained from the restrict-and-append algorithm and the upper bound obtained from the cut-based decomposition algorithm. In the column labeled “RelativeGap,” we report the average of relative gaps that are greater than 10^{-4} and the number of corresponding instances out of ten in parentheses. We use label ZRG—zero relative gap—when the relative gaps are less than 10^{-4} for all instances.

From the results in Table 3.2, we observe that the number of scenarios significantly impacts the performance of the reformulation-based algorithm, even for the relatively smaller-sized instances (N40 and N60), because the size of the reformulation (3.6) increases with the increase in the number of scenarios. In contrast, the cut-based decomposition approach is

Table 3.2: Computational results for solving DRR-SPIP US20 instances using reformulation-based algorithm, cut-based decomposition algorithm, and restrict-and-append algorithm.

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

395 times (on average) faster for the instances solved by both the algorithms. Interestingly, the restrict-and-append algorithm produces equal or better solutions than the cut-based decomposition algorithm for 541 out of 600 (around 90 %) instances, and the former is 4.2 times (on average) faster than the latter in solving the instances that the cut-based decomposition algorithm solves to optimality. Note that the restrict-and-append algorithm does not guarantee to provide an optimal solution, and we are able to evaluate the quality of its solution only because the cut-based decomposition algorithm provides an optimal solution in finite iterations.

Table 3.3: Computational results for solving DRR-SPIP US100 instances using reformulation-based algorithm, cut-based decomposition algorithm, and restrict-and-append algorithm.

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

In Table 3.3, we report the computational results for solving the DRR-SPIP US100 instances using the proposed algorithms. The cut-based decomposition algorithm is 34 times (on average) faster than the reformulation-based algorithm for the optimally solved instances. Moreover, for the unsolved instances, the former provides the optimality gap that is 38% (on average) of the optimality gap reported by the latter. We can observe that even the cut-based decomposition algorithm cannot solve around 84 % (201 out of 240) of the larger-sized instances in N80 and N100 due to the higher level of uncertainty than the US20 instances. This is because generating an optimality cut is computationally more expensive for the

US100 instances, since the algorithm has to compute the cut coefficients $\beta_a(x_a)$ for all arcs $a \in A$ that have uncertainty. Similar to the results for the US20 instances, the restrict-and-append algorithm produces equal or better solutions than the cut-based decomposition algorithm for 484 out of 600 (around 81 %) instances. For the instances that the cut-based decomposition algorithm solves to optimality, the restrict-and-append algorithm is 22 times faster on average; whereas for the remaining instances, the restrict-and-append algorithm takes 1539 seconds (on average).

3.5.3 Computational Results for DRO-SPIP

In Table 3.4 and 3.5, we present the computational results obtained by solving the DRO-SPIP instances. Labels “Decomposition” and “RelaxAppend” in the table denote the decomposition algorithm and the relax-and-append algorithm, respectively. In each row, we report the average value of the results for ten instances. The definitions of labels T , Gap, ZG, 3600+ are the same as presented in the previous section. Note that we define the optimality gap for the relax-and-append algorithm as $(z_{RO}^{ub,L} - z_{RO}^{lb,L})/z_{RO}^{lb,L}$ where L is the last iteration that solves the relaxation of DRO-SPIP to optimality. In case the relax-and-append algorithm reaches the time limit before terminating the branch-and-cut algorithm for the first iteration, we get $z_{RO}^{ub,1}$ from the upper bound of the branch-and-cut algorithm and $z_{RO}^{lb,1}$ by solving the outer distribution separation problem using the best incumbent solution. For the relax-and-append algorithm, we additionally report the average number of iterations until the termination under label “#Iter.”

According to Table 3.4, the decomposition algorithm is 1.2 times (on average) faster than the relax-and-append algorithm in solving the instances to optimality. The decomposition algorithm solved all 600 instances, except one, within 63 seconds (on average), whereas

Table 3.4: Computational results for solving DRO-SPIP US20 instances using decomposition algorithm and relax-and-append algorithm.

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

the relax-and-append algorithm could not solve four instances to optimality. Also, the optimality gaps for the unsolved instances are less than 0.1 % for both algorithms. The relax-and-append algorithm requires 1.5 iterations (on average), which implies some instances are solved immediately after the first iteration of the algorithm. We observe that for the US20 instances, the relax-and-append algorithm terminates after the first iteration when the interdicator's optimal decision is to interdict the arcs with no parameter uncertainty, so that any probability distribution leads to the same objective value. Note that the relax-and-

Table 3.5: Computational results for solving DRO-SPIP US100 instances using decomposition algorithm and relax-and-append algorithm.

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

append algorithm utilizes the explicit enumeration for the distribution separation instead of solving a linear program as done in the decomposition algorithm. For a small-sized subset of ambiguity set, this significantly reduces the time required for solving the distribution separation problem, and therefore, the relax-and-append algorithm is likely to be faster than the decomposition algorithm when it terminates within a small number of iterations.

In Table 3.5, we observe that the relax-and-append algorithm requires 5.7 iterations on average because of the higher level of uncertainty in the US100 instances than the US20 instances.

This results in a larger performance difference between the algorithms; the decomposition approach is 2.5 times (on average) faster than the relax-and-append algorithm.

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

Lastly, we discuss the impact of the choice of ambiguity set on the solution time. For DRO-SPIP with the moment matching set, the size of the distribution separation problem is highly dependent on the number of arcs associated with uncertain parameters since it impacts the number of constraints. This causes the proposed algorithms to spend more time in solving the distribution separation problem for the US100 instances than the US20 instances. To show this, we compute the ratio of the time for solving a distribution separation problem to the total solution time taken for solving each of the US20 and US100 instances using the decomposition algorithm for DRO-SPIP. The average of the ratios for the US20 instances is computed to be 9.1%, while the same for the US100 instances is 67.7%. On the other hand, for the Wasserstein ambiguity set, the average of the ratios for the US20 instances and the US100 instances are 20% and 29.2%, respectively. This is because the number of arcs having uncertainty does not impact the number of variables and constraints in the Wasserstein ambiguity set (3.18); instead, it is largely impacted by the number of scenarios $|\Omega|$. For example, the average of the ratios for US100 instances for $|\Omega| = 100, 500, \text{ and } 1000$ are 3.6%, 21%, and 36.8%, respectively.

For DRR-SPIP, we also consider the time spent in computing cut coefficients (3.8), which

involves solving the corresponding distribution problem. Similar to DRO-SPIP, we observe that the average of the time taken in solving distribution separation problems and in computing the cut coefficients for the cut-based decomposition algorithm is largely impacted by the number of arcs having uncertainty with the moment-matching set and by the number of scenarios with the Wasserstein ambiguity set.

3.5.4 Comparison between Risk-Neutral, Distributionally Robust, and Distributionally Risk-Receptive Solutions

In this section, we evaluate the solutions obtained from DRO-SPIP, DRR-SPIP, and (risk-neutral) S-SPIP (3.1). For S-SPIP instances, we assume that each scenario is realized with equal probability. For test instances, we use the US100 instances so that there is no trivial optimal solution that interdicts the arcs having no uncertainty. Also, we fix the number of scenarios to 100, as we observe in our preliminary tests that the number of scenarios has only a minor impact on the difference among distributionally risk-receptive, distributionally robust, and risk-neutral solution values. We plot and compare optimal objective values for different risk preference in Figures 3.2, 3.3, and 3.4.

In Figure 3.2, we present the optimal objective values for ten instances from N40 category with $b = 2, 5, \text{ and } 10$. Comparing three graphs 3.2a-3.2c, we can see that the gap between risk-receptive and robust solution values grows as b increases. Next, in Figures 3.3 and 3.4, we present the optimal objective values for N60 instances with the moment matching set and N80 instances with the Wasserstein set, respectively. Given the moments of random variables $u_q, q = 1, \dots, m$, the size of the moment matching set increases as ϵ_M increases. Likewise, the Wasserstein ambiguity set's size increases as the Wasserstein distance limit (ϵ_W) increases. In Figures 3.3 and 3.4, we can see that the gap between the robust and risk-receptive solution

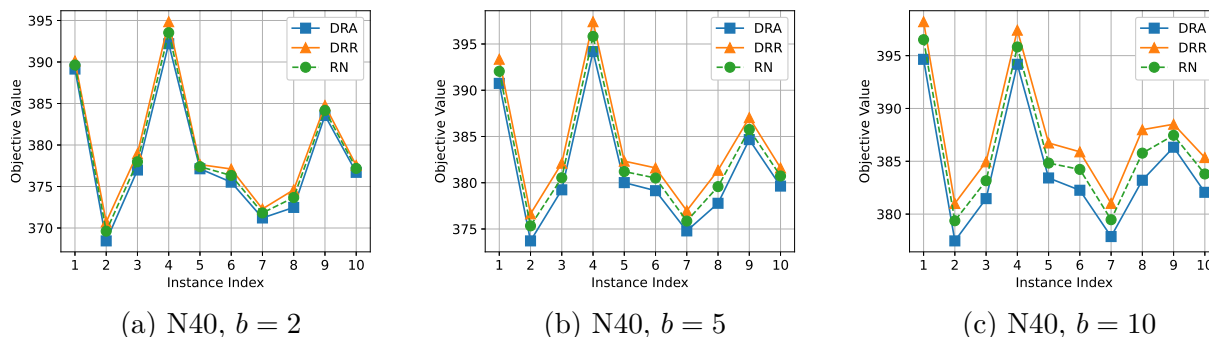


Figure 3.2: Optimal objective values for instances of N40, US100, obtained by solving DRO-SPIP (DRA), DRR-SPIP (DRR), and S-SPIP (RN) with Wasserstein ambiguity set and $b \in \{2, 5, 10\}$.

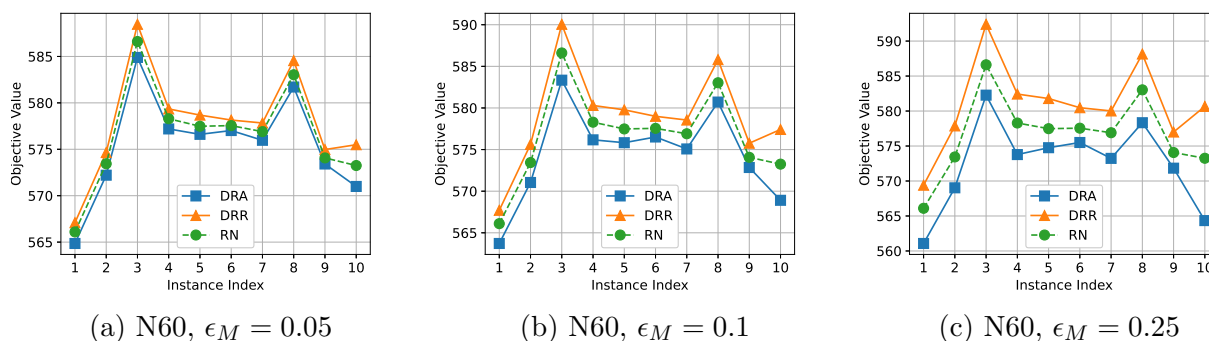


Figure 3.3: Optimal objective values for instances of N60, US100, obtained by solving DRO-SPIP (DRA), DRR-SPIP (DRR), and S-SPIP (RN) with moment matching set defined using $\epsilon_M \in \{0.05, 0.1, 0.25\}$.

values becomes larger as we increase ϵ_M and the multiplication factor ρ of ϵ_W . These results show that the outcome under distributional uncertainty may vary significantly when the interdicator is capable of interdicting many arcs, and the decision makers are less confident about the ambiguity set (in other words, have a larger ambiguity set). However, in such a case, the DRR-SPIP and DRO-SPIP frameworks provide optimal solutions for the worst-case and best-case probability distributions, which establish the bounds on the outcome of S-SPIP, respectively, that can help the decision makers in conducting the worst-case and best-case analysis.

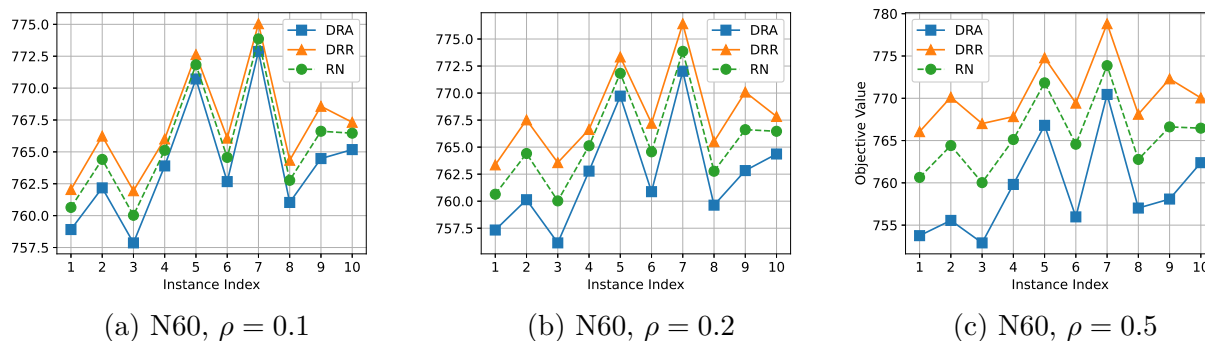


Figure 3.4: Optimal objective values for instances of N80, US100, obtained by solving DRO-SPIP (DRA), DRR-SPIP (DRR), and S-SPIP (RN) with Wasserstein ambiguity set defined using $\rho \in \{0.1, 0.2, 0.5\}$.

From the perspective of the network user, an optimal risk-receptive solution gives information of the most vulnerable arcs, whose failure can significantly increase the minimum-cost path. Therefore, DRR-SPIP is an appropriate model for the network user in terms of vulnerability analysis. On the other hand, for the interdictor, the DRR-SPIP and DRO-SPIP frameworks provide a range of possible outcomes of interdiction attempts. This range provides the interdictor with the comparative analysis of the expected outcome according to the risk appetite of the decision maker. Suppose the situation where the interdictor has relatively small budget or high confidence on the distributional uncertainty like in Figures 3.2a, 3.3a and 3.4a. Then, the gap between robust and risk-receptive objective values will be minor, and the interdictor can realize that there is no incentive to choose an optimal risk-receptive interdiction strategy in comparison to an optimal robust strategy. Hence, both DRR-SPIP and DRO-SPIP can be useful from the interdictor's perspective.

3.6 Conclusion

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

Chapter 4.

Distributionally Risk-Receptive and Robust Multistage Stochastic Integer Programs and Interdiction Models¹

4.1 Introduction

In this chapter, we extend the DRR and DRO frameworks to multistage stochastic programming, in which a sequence of decisions is made over a time horizon. At each stage, a multistage stochastic program aims to minimize the sum of the current cost and the expected costs of future stages. We study these models in various settings, including integer programming, disjunctive programming, and both decision-independent and decision-dependent uncertainty.

For the decision-independent uncertainty case, we address DRR and DRO multistage stochastic integer programs (DRR- and DRO-MSIPs) by using convex lower-approximations of the expected cost-to-go functions for best-case or worst-case distributions. We derive valid cutting planes and utilize reformulation techniques, considering finite or continuous support. We note that while the reformulation-based approximations are derived from standard techniques, they are presented in this chapter for comparative purposes, as no other approaches

¹Reference [51]: Sumin Kang and Manish Bansal. Distributionally risk-receptive and robust multi-stage stochastic integer programs and interdiction models. *Mathematical Programming*, 2025.

for solving DRR-MSIPs are known in the literature. These approximations enable the application of decomposition methods, which effectively reduce complexity in our multistage decision-making setting. Specifically, we propose SDDP (Stochastic Dual Dynamic Programming [77]) algorithms tailored for DRR- and DRO-MSIPs that utilize these approximations. We also generalize these algorithms to solve DRR and DRO multistage stochastic disjunctive linear programs (DRR- and DRO-MSDPs) where feasible sets are defined by disjunctive linear sets.

In the literature, DRO models with decision-dependent ambiguity sets have been addressed using their reformulations (see, e.g., [7, 60, 106]). However, to the best of our knowledge, a solution approach for DRR (or optimistic) models with decision-dependent ambiguity sets has not been studied. We fill this gap by introducing a reformulation-based approximation for decision-dependent DRR-MSIPs where the ambiguity sets are defined using the Wasserstein distance. We also present lower-bound approximation for the associated cost-to-go function and then utilize it to introduce a cutting plane-based algorithm for DRR-MSIPs with decision-dependent uncertainty.

To demonstrate practical applications of DRR- and DRO-MSIPs, we consider multistage NIPs. Specifically, we introduce multistage stochastic models for the maximum flow interdiction problem and the facility location interdiction problem, denoted by MS-MFIP and MS-FLIP, respectively. MS-MFIP has applications in disrupting domestic sex trafficking networks [53] and illegal drug supply chain networks [62], while MS-FLIP is applicable for identifying critical assets in infrastructure systems [24]. Despite these applications, their multistage stochastic variants have not been studied in the literature. To the best of our knowledge, only two-stage variants have been explored in [26, 47, 85], where each stage corresponds to the decision-making process of a single player.

Through our DRO and DRR frameworks, we investigate the significance of varying risk

appetite of the decision-maker in MS-MFIP. Also, we examine an adversarial setting where data are corrupted by malicious injections and demonstrate that the DRR framework can enhance the out-of-sample performance in such environments. Lastly, we provide numerical results for both MS-MFIP and MS-FLIP, showcasing the computational efficiency of the proposed approaches.

In the remainder of this section, we present formulations of general DRR- and DRO multistage stochastic programs (MSPs) along with the contributions and organization of this chapter.

4.1.1 Problem Formulation: DRO-MSP and DRR-MSP

We first present the Bellman equation of an (risk-neutral) MSP with a planning horizon of T stages and then generalize it to the cases with distributional ambiguity, i.e., DRO- and DRR-MSPs. There are two types of decision variables at each stage: *state* decision variables that influence subsequent stages and *local* decision variables specific to the current stage. Let x_t and y_t be the state and local decision vectors for stage $t \in [T]$, respectively. An MSP is formulated as

$$\min_{(x_1, y_1) \in X_1} \left\{ f_1(x_1, y_1) + \mathbb{E}_{P_2} [Q_2(x_1, \omega_2)] \right\}, \quad (4.1)$$

where the cost-to-go functions are defined as

$$Q_t(x_{t-1}, \omega_t) = \min_{(x_t, y_t) \in X_t(x_{t-1}, \omega_t)} \left\{ f_t(x_t, y_t, \omega_t) + \mathbb{E}_{P_{t+1}} [Q_{t+1}(x_t, \omega_{t+1})] \right\}, \quad (4.2)$$

for $t = 2, \dots, T$, and $\omega_t \in \Omega_t$. Here, let the function $Q_{T+1} = 0$. Set X_1 denotes the feasible region of the first stage, and set $X_t(x_{t-1}, \omega_t)$ denotes the feasible region of each stage $t \in \{2, \dots, T\}$ that depends on a decision of the previous stage, x_{t-1} . The random vector is

denoted by ξ_t and is realized for each scenario $\omega_t \in \Omega_t$ with probability $P_t(\omega_t)$. We denote the probability distribution at stage t by P_t for $t = 2, \dots, T$.

Now, to present DRO- and DRR-MSPs, we consider a set \mathcal{P}_t of probability distributions as the ambiguity set for $t \in \{2, \dots, T\}$. Then, the bellman equation form of a DRO-MSP is given by

$$\min_{(x_1, y_1) \in X_1} \left\{ f_1(x_1, y_1) + \max_{P_2 \in \mathcal{P}_2} \mathbb{E}_{P_2} [Q_2^{RA}(x_1, \omega_2)] \right\}, \quad (4.3)$$

where

$$Q_t^{RA}(x_{t-1}, \omega_t) = \min_{(x_t, y_t) \in X_t(x_{t-1}, \omega_t)} \left\{ f_t(x_t, y_t, \omega_t) + \max_{P_{t+1} \in \mathcal{P}_{t+1}} \mathbb{E}_{P_{t+1}} [Q_{t+1}^{RA}(x_t, \omega_{t+1})] \right\} \quad (4.4)$$

for $t = 2, \dots, T$ and $\omega_t \in \Omega_t$. Here, let $Q_{T+1}^{RA} = 0$. We refer to the value function

$$Q_{t+1}^{RA}(x_t) := \max_{P_{t+1} \in \mathcal{P}_{t+1}} \mathbb{E}_{P_{t+1}} [Q_{t+1}^{RA}(x_t, \omega_{t+1})]$$

as the *pessimistic expected cost-to-go* function. By minimizing with respect to the probability distribution within the ambiguity set in (4.3) and (4.4), instead of maximizing, we can formulate a DRR-MSP as follows:

$$\min_{(x_1, y_1) \in X_1} \left\{ f_1(x_1, y_1) + \min_{P_2 \in \mathcal{P}_2} \mathbb{E}_{P_2} [Q_2^{RR}(x_1, \omega_2)] \right\}, \quad (4.5)$$

where

$$Q_t^{RR}(x_{t-1}, \omega_t) = \min_{(x_t, y_t) \in X_t(x_{t-1}, \omega_t)} \left\{ f_t(x_t, y_t, \omega_t) + \min_{P_{t+1} \in \mathcal{P}_{t+1}} \mathbb{E}_{P_{t+1}} [Q_{t+1}^{RR}(x_t, \omega_{t+1})] \right\} \quad (4.6)$$

for $t = 2, \dots, T$ and $\omega_t \in \Omega_t$. Let $Q_{T+1}^{RR} = 0$. The value function

$$Q_{t+1}^{RR}(x_t) := \min_{P_{t+1} \in \mathcal{P}_{t+1}} \mathbb{E}_{P_{t+1}} [Q_{t+1}^{RR}(x_t, \omega_{t+1})]$$

is referred to as the *optimistic expected cost-to-go* function. For DRR-MSPs with decision-dependent uncertainty, we consider ambiguity sets that depend on decisions, i.e., the optimistic expected cost-to-go functions are defined by

$$Q_{t+1}^{RR}(x_t) = \min_{P_{t+1} \in \mathcal{P}_{t+1}(x_t)} \mathbb{E}_{P_{t+1}} [Q_{t+1}^{RR}(x_t, \omega_{t+1})].$$

When the feasible region $X_t(x_{t-1}, \omega_t)$ is defined by a set of linear inequalities and disjunctive constraints, problems (4.3)-(4.4) and (4.5)-(4.6) are referred to as DRO- and DRR-MSDPs, respectively, i.e., for $\omega_t \in \Omega_t$,

$$X_t(x_{t-1}, \omega_t) := \left\{ (x_t, y_t) \in \mathbb{R}_+^{d_x} \times \mathbb{R}_+^{d_y} : \bigvee_{h \in H_t} \left(A_t^h(\omega_t)x_t + B_t^h(\omega_t)y_t \geq b_t^h(\omega_t) - C_t^h(\omega_t)x_{t-1} \right) \right\}. \quad (4.7)$$

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 (4.7) represent linear inequalities with integrality constraints on variables, then the DRO- and DRR-MSDPs reduce to DRO- and DRR-MSIPs. The main challenges encountered in solving these problems arise 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.

The organization of the remainder of this chapter and our contributions are as follows.

- *DRR-MSIPs with finite support.* In Section 4.2, we present a class of cutting planes to under-approximate the optimistic expected cost-to-go function Q_{t+1}^{RR} at stage t . For the sake of computational comparisons, we also reformulate DRR-MSIPs by considering the probability distribution at each stage as a decision vector and applying a McCormick envelope. We show that our SDDP-based algorithms using the cutting plane- and reformulation-based approximations are exact and finitely convergent (Section 4.5). Our computational results in Section 4.7 show that the algorithm using the cutting plane-based approach is 24.1 times faster (on average) than the algorithm using the reformulation approach.
- *DRR-MSIPs with continuous support and decision-dependent ambiguity sets.* In Section 4.3, we study DRR-MSIPs with Wasserstein ambiguity sets where the radii of the Wasserstein balls depend on the decision variables. We derive a dual formulation of the inner minimization at each stage (4.6), thereby deriving a class of valid cutting planes for the optimistic expected cost-to-go function and a cutting plane-based solution approach.
- *DRO-MSIPs.* In Section 4.4, we present a cutting plane-based approximation for DRO-MSIPs with a general family of ambiguity sets and finite supports. Also, we present a reformulation-based approximation using a dual formulation of DRO-MSIPs with Wasserstein ambiguity sets and continuous/finite support. We observe that the former is 28 times (on average) faster than the latter approach for solving DRO-MSIP with finite support.
- *Out-of-sample Performance.* By conducting out-of-sample tests, we demonstrate the significance of DRR and DRO in the context of interdiction problem, in particular,

MS-MFIP. The results in Section 4.7.1.2 show that the DRO framework enhances the robustness of decision policies under uncertainty, and the DRR framework enables the identification of network vulnerabilities by giving more weights to unfavorable scenarios for the network user. Furthermore, in Section 4.7.1.3 we illustrate out-of-sample performance in an adversarial setting, where the sample data are intentionally corrupted (refer to robust statistics in Section 2.1.5 for details). The results show that the DRR policies behave robustly to data corruption by decreasing the significance of the corrupted data points.

- *DRR-MSDPs and DRO-MSDPs.* In Section 4.6, we introduce algorithms for solving MSDPs, DRR-MSDPs, and DRO-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 4.1, MS-MFIP and MS-FLIP are important interdiction problems in their own right and have not been studied in the literature. We present algorithms to solve these problems and their distributionally ambiguous variants, thereby generalizing results of [24, 26, 47, 53, 62] that study special cases of these problems.

In Section 4.7, we present our computational results and concluding remarks in Section 4.8.

Throughout this chapter, we made the following assumptions:

Assumption 4.1. The random vectors are stage-wise independent, i.e., $\boldsymbol{\xi}_t$ is independent of $\boldsymbol{\xi}_{[t-1]} = (\boldsymbol{\xi}_2, \dots, \boldsymbol{\xi}_{t-1})$ for all $t = 3, \dots, T$.

We note that the stage-wise independence assumption is required for computationally efficient algorithms but not for deriving convex approximations.

Assumption 4.2. The state variables x_t are binary for all $t \in [T]$. The feasible sets X_1 and $X_t(\cdot, \cdot)$ are defined as mixed-integer sets (and disjunctive sets only in Section 4.6).

Remark 4.1. For general integer or discrete state variables x_t that are bounded, we can obtain their equivalent binary representation by using the binary expansion.

Assumption 4.3. 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 $t = 2, \dots, T$, are nonempty and compact.

Assumption 4.4. For each $t = 2, \dots, T$, the support of ξ_t is finite, i.e., $|\Omega_t| < \infty$. Accordingly, let p_t^i be the probability of scenario ω_t^i , i.e., $P_t(\omega_t^i)$, for $i \in \mathcal{N}_t$, where $\mathcal{N}_t := \{1, \dots, N_t\}$ is the index set associated with Ω_t , $t = 2, \dots, T$.

Throughout this chapter, we assume that ξ_t has finite support unless stated otherwise. Specifically, we will relax this assumption and consider continuous supports in Section 4.3 and Remark 4.2 in Section 4.4.

Assumption 4.5. Functions $f_1(x_1, y_1)$ and $f_t(x_t, y_t, \omega_t)$, for $\omega_t \in \Omega_t$ and $t = 2, \dots, T$ are linear.

4.2 Convex Approximations for DRR-MSIPs having Finite Supports

In this section, we present two convex approximations of the optimistic expected cost-to-go function $Q_{t+1}^{RR}(x_t)$ at stage t . We start by considering affine cuts valid for Q_{t+1}^{RR} :

$$Q_{t+1}^{RR}(x_t, \omega_{t+1}^i) \geq (\alpha_t^{i,k})^\top x_t + \beta_t^{i,k}, \quad \forall x_t \in \{0, 1\}^{d_x}, \quad (4.8)$$

and denote them by their coefficients $(\alpha_t^{i,k}, \beta_t^{i,k})$, where $i \in \mathcal{N}_{t+1}$ indexes scenarios in Ω_{t+1} , and $k \in [K_t]$, with $K_t \geq 1$ represents the number of cuts. Throughout this section, we suppose these cuts are given for every stage t . Discussions regarding the cut generating procedure and properties of these cuts are deferred to Section 4.5. Using the cuts, we define the following nonconvex (bilinear) approximating problem:

$$\phi_t^B(x_t) := \min_{\theta_t, P_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i \theta_t^i \quad (4.9a)$$

$$\text{s.t. } \theta_t^i \geq (\alpha_t^{i,k})^\top x_t + \beta_t^{i,k}, \quad \forall k \in [K_t], i \in \mathcal{N}_{t+1}, \quad (4.9b)$$

$$P_{t+1} \in \mathcal{P}_{t+1}. \quad (4.9c)$$

Note that we consider the probabilities $P_{t+1} = (p_{t+1}^i)_{i \in \mathcal{N}_{t+1}}$ as decision variables. This leads to bilinear terms involving variables p_{t+1}^i and θ_t^i in the objective function. By construction, the function $\phi_t^B(x_t)$ provides a lower bound for the optimistic expected cost-to-go function $Q_{t+1}^{RR}(x_t)$, yet solving this problem directly is not desirable due to the bilinear terms. In the following sections, we derive convex approximations of this problem using a new class of cutting planes and a mixed-integer linear programming reformulation, respectively. For simplicity of exposition, we let $K_t = 1$ for all $t \in [T-1]$ and suppress the index k in notation.

4.2.1 A Cutting Plane-Based Approximation for DRR-MSIP

Consider any $\hat{x}_t \in \{0, 1\}^{d_x}$. We define a cutting plane-based approximating function as:

$$\phi_t^C(x_t) := \min \left\{ \phi : \phi \geq \pi_t^\top (x_t - \hat{x}_t) + \gamma_t \right\} \quad (4.10)$$

where the parameters of the inequalities are given by

$$\pi_{t,j} := \begin{cases} \min_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i \alpha_{t,j}^i, & \text{if } \hat{x}_{t,j} = 0, \\ \max_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i \alpha_{t,j}^i, & \text{if } \hat{x}_{t,j} = 1, \end{cases} \quad \text{for } j \in [d_x], \quad (4.11)$$

and $\gamma_t := \min_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i \left((\alpha_t^i)^\top \hat{x}_t + \beta_t^i \right)$, for each $t \in [T-1]$. Apparently, this function is convex. Also, it provides a lower-bound for \mathcal{Q}_{t+1}^{RR} as shown in Theorem 4.1.

Theorem 4.1. *The function ϕ_t^C provides a lower bound for the optimistic expected cost-to-go function, i.e., $\phi_t^C(x_t) \leq \mathcal{Q}_{t+1}^{RR}(x_t)$ for all $x_t \in \{0, 1\}^{d_x}$ and $t \in [T-1]$.*

An important property of this cut is that it preserves the tightness of the cuts (α_t^i, β_t^i) , i.e., it intersects with \mathcal{Q}_{t+1}^{RR} at \hat{x}_t if the cuts (α_t^i, β_t^i) intersect with $\mathcal{Q}_{t+1}^{RR}(\cdot, \omega_{t+1}^i)$ at \hat{x}_t for all $i \in \mathcal{N}_{t+1}$. This property plays a key role in showing the finite convergence of our algorithm presented in Section 4.5. Additionally, it is important to note that the cuts (4.10) are different from Benders cuts, which rely on the duality results of problems. Since the DRR problem (4.6) is nonconvex even when involving continuous variables, a convex combination of Benders cuts cannot be applied to approximate the optimistic expected cost-to-go function $\mathcal{Q}_{t+1}^{RR}(x_t)$ as it can be done for the risk-neutral or pessimistic expected cost-to-go-function.

4.2.2 A Reformulation-Based Approximation for DRR-MSIP

We present a reformulation-based approximation for DRR-MSIP by treating the probability distribution as variables and using McCormick envelopes for bilinear terms. Though straightforward, it remains the only approach using existing techniques to solve this problem and therefore, we include it for comparative purpose. More specifically, it works as follows.

For each constraint in (4.9a), we multiply p_{t+1}^i to both sides of inequalities for each $i \in \mathcal{N}_{t+1}$, and replace $p_{t+1}^i x_t$ with a decision vector η_t^i in the right-hand side of the resulting inequalities. This yields the following system of inequalities:

$$p_{t+1}^i \theta_t^i \geq (\alpha_t^i)^\top \eta_t^i + \beta_t^i p_{t+1}^i, \quad \forall i \in \mathcal{N}_{t+1}, \quad (4.12a)$$

$$\eta_t^i \leq x_t, \quad \eta_t^i \leq p_{t+1}^i, \quad \eta_t^i \geq p_{t+1}^i + x_t - 1, \quad \eta_t^i \geq 0, \quad \forall i \in \mathcal{N}_{t+1}. \quad (4.12b)$$

Notice that a system of inequalities (4.12b) ensure that a feasible η_t^i equals to $p_{t+1}^i x_t$, given any $x_t \in \{0, 1\}^{d_x}$ and $(p_{t+1}^i)_{i \in \mathcal{N}_{t+1}} \in [0, 1]^{N_{t+1}}$. We introduce an additional variable $\bar{\theta}_t^i$ to replace $p_{t+1}^i \theta_t^i$. This yields an approximating function of \mathcal{Q}_{t+1}^{RR} for each $t \in [T - 1]$ as follows:

$$\phi_t^R(x_t) := \min_{P_{t+1}, \bar{\theta}, \eta_t} \left\{ \sum_{i \in \mathcal{N}_{t+1}} \bar{\theta}_i : \bar{\theta}_i \geq (\alpha_t^i)^\top \eta_t^i + \beta_t^i p_{t+1}^i, \quad \forall i \in \mathcal{N}_{t+1}, \quad (4.13a) \right.$$

$$\left. (4.12b), P_{t+1} \in \mathcal{P}_{t+1} \right\}. \quad (4.13b)$$

Since θ_t^i is not restricted, we can readily show that the equivalence between (4.12a) and constraint (4.13a). Therefore, the function ϕ_t^R equals to the function ϕ_t^B , which is a lower bound for $\mathcal{Q}_{t+1}^{RR}(x_t)$. When the ambiguity set \mathcal{P}_{t+1} is defined by a polytope (e.g., Wasserstein ambiguity set with a finite support), the problem (4.13) is a linear program. Consequently,

ϕ_t^R becomes piecewise linear and convex by linear programming duality.

4.3 Convex Approximation for DRR-MSIPs having Continuous Supports and Decision-Dependent Ambiguity Sets

In this section, we investigate DRR-MSIPs, where for each stage t the support of ξ_t is continuous. Suppose that a finite set of data $\bar{\Omega}_t := \{\omega_t^1, \dots, \omega_t^{N_t}\}$ is available. Consider an empirical distribution $\bar{P}_t = \frac{1}{N_t} \sum_{i \in \mathcal{N}_t} \delta_{\omega_t^i}$, where $\delta_{\omega_t^i}$ is the Dirac delta function centered at ω_t^i for $i \in \mathcal{N}_t$. We now present a dual-based convex approximation of the optimistic expected cost-to-go function $\mathcal{Q}_{t+1}^{RR}(x_t)$ at stage t . This approximation can be applied to a further generalized case where ambiguity sets are decision-dependent. In particular, we focus on decision-dependent ambiguity sets defined using Wasserstein metric as follows:

$$\mathcal{P}_t(x_{t-1}) := \left\{ P_t \in \mathcal{M}(\Omega_t) : \mathcal{W}(P_t, \bar{P}_t) \leq \epsilon_t(x_{t-1}) \right\} \quad (4.14)$$

for $t = 2, \dots, T$ [60], where $\mathcal{M}(\Omega_t)$ is a set of all probability distributions supported on Ω_t , and $\mathcal{W}(P_t, \bar{P}_t)$ is the 1-Wasserstein distance defined as

$$\mathcal{W}(P_t, \bar{P}_t) := \inf_{P \in \mathcal{P}(\Omega_t \times \Omega_t)} \left\{ \mathbb{E}_P \left[\|\xi_t(\omega_t) - \xi_t(\omega'_t)\| \right] : P(\omega_t) = P_t, P(\omega'_t) = \bar{P}_t \right\}. \quad (4.15)$$

Here, $\mathcal{P}(\Omega_t \times \Omega_t)$ is the set of all joint probability distributions supported on $\Omega_t \times \Omega_t$. The marginal distributions of ω_t and ω'_t are denoted by $P(\omega_t)$ and $P(\omega'_t)$, respectively, and $\|\cdot\|$ is an arbitrary norm. For a given $\epsilon_t(x_{t-1}) > 0$, each ambiguity set $\mathcal{P}_t(x_{t-1}), t \in \{2, \dots, T\}$, is a Wasserstein ball containing all probability distributions within a certain radius $\epsilon_t(x_{t-1})$

from the empirical probability distribution. For the ease of exposition in this section, we use x_t to represent all decision variables at each stage t .

Proposition 4.2 (Strong duality). *For the ambiguity set $\mathcal{P}_t(x_{t-1})$ defined as (4.14), the optimistic expected cost-to-go function can be reformulated as*

$$\begin{aligned} Q_t^{RR}(x_{t-1}) &= \min_{P_t \in \mathcal{P}_t(x_{t-1})} \mathbb{E}[Q_t^{RR}(x_{t-1}, \omega_t)] \\ &= \max_{\rho_t \geq 0} \left\{ -\epsilon_t(x_{t-1})\rho_t + \sum_{i \in \mathcal{N}_t} \frac{1}{N_t} \min_{\omega_t \in \Omega_t} \left\{ \rho_t \|\xi_t(\omega_t) - \xi_t(\omega_t^i)\| + Q_t^{RR}(x_{t-1}, \omega_t) \right\} \right\}. \end{aligned} \quad (4.16)$$

Using Proposition 4.2, we derive an approximation for Q_t^{RR} in Theorem 4.3 under the following assumptions.

Assumption 4.6. Random vector $\xi_t \in \mathbb{R}^{d_\omega}$ is only associated with the right-hand side of the constraints defining $X_t(x_{t-1}, \omega_t)$ for each stage t . That is, the feasible set for each $\omega_t \in \Omega_t$ is written as

$$X_t(x_{t-1}, \omega_t) = \left\{ x_t \in \mathcal{X}_t : A_t x_t \geq \omega_t - C_t x_{t-1} \right\}, \quad (4.17)$$

where $\mathcal{X}_t \subseteq \mathbb{R}^{d_x}$ is the set defined by the integrality constraints for x_t . For simplicity (with a slight abuse of notation), we let ω_t denote the realization $\xi_t(\omega_t)$ only in this section. Additionally, we use Ω_t to denote the support of ξ_t .

Assumption 4.7. The support of ξ_t is a bounded hyperrectangle, i.e., $\Omega_t = [l_t, u_t]^{d_\omega}$, where $l_t, u_t \in \mathbb{R}$ and $l_t \leq u_t$, for each stage t .

Assumption 4.8. The decision-dependent radius $\epsilon_t(x_{t-1}) : \mathcal{X}_{t-1} \rightarrow \mathbb{R}_+$ is an affine function for each stage t .

Theorem 4.3. *Given valid cuts' coefficients $\{(\pi_t^k, \gamma_t^k)\}_{k \in [K_t]}$ for $Q_{t+1}^{RR}(x_t)$,*

(a) the optimistic expected cost-to-go function $\mathcal{Q}_t^{RR}(x_{t-1})$ is lower-approximated as follows:

$$\begin{aligned} \mathcal{Q}_t^{RR}(x_{t-1}) \geq & -\epsilon_t(x_{t-1})\rho_t + \sum_{i \in \mathcal{N}_t} \frac{1}{N_t} \left((-\lambda_t^i)^\top C_t x_{t-1} + (\mu_t^i)^\top l_t \right. \\ & \left. - (\nu_t^i)^\top u_t + (\lambda_t^i - \mu_t^i + \nu_t^i)^\top \omega_t^i + \sum_{k \in [K_t]} \zeta_{tk}^i \gamma_t^k \right), \end{aligned} \quad (4.18)$$

such that parameters $(\rho_t, \{\lambda_t^i, \mu_t^i, \nu_t^i, \zeta_t^i\}_{i \in \mathcal{N}_t})$ satisfy the following system of equations and inequalities (4.19):

$$A_t^\top \lambda_t^i - \sum_{k \in [K_t]} \zeta_{tk}^i \pi_t^k = c_t, \quad \forall i \in \mathcal{N}_t, \quad (4.19a)$$

$$\|\lambda_t^i + \mu_t^i - \nu_t^i\|_* \leq \rho_t, \quad \forall i \in \mathcal{N}_t, \quad (4.19b)$$

$$\sum_{k \in [K_t]} \zeta_{tk}^i = 1, \quad \forall i \in \mathcal{N}_t, \quad (4.19c)$$

$$\rho_t \geq 0, \quad (\lambda_t^i, \mu_t^i, \nu_t^i, \zeta_t^i) \geq 0, \quad \forall i \in \mathcal{N}_t, \quad (4.19d)$$

where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$.

(b) the strongest lower bound approximation (4.18) is obtained by solving the following problem:

$$\begin{aligned} \max \left\{ & -\epsilon_t(x_{t-1})\rho_t + \frac{1}{N_t} \sum_{i \in \mathcal{N}_t} \left((-\lambda_t^i)^\top C_t x_{t-1} + (\mu_t^i)^\top l_t \right. \right. \\ & \left. \left. - (\nu_t^i)^\top u_t + (\lambda_t^i - \mu_t^i + \nu_t^i)^\top \omega_t^i + \sum_{k \in [K_t]} \zeta_{tk}^i \gamma_t^k \right) : (4.19a)-(4.19d) \right\}. \end{aligned} \quad (4.20)$$

Problem (4.20) is a cut-generating problem, where any feasible solution yields a cut in the form of (4.18). With the choice of specific norms, problem (4.20) becomes computationally manageable. For instance, if $\|\cdot\|$ represents the l_1 norm, the dual norm is the l_∞ norm, allowing the linearization of constraints (4.19b) and the transformation of problem (4.20)

into a linear program. Also, if $\|\cdot\|$ is the l_2 norm, its dual norm is also the l_2 norm, resulting in problem (4.20) becoming a second-order conic program.

Using the cuts generated as above, we derive a cutting plane-based approximating function for Q_t^{RR} as in the form of (4.10). It is important to note that the cuts for next stage ($t + 1$) in Theorem 4.3 are naturally given through our algorithmic procedure (discussed in Section 4.5). Our algorithm produces cuts in a descending order of stages, from T down to 1, and thus there always exists cuts for stage ($t + 1$) when generating a cut at stage t .

4.4 Convex Approximations for DRO-MSIPS

In this section, we present two convex approximations of the pessimistic expected cost-to-go functions $Q_{t+1}^{RA}(x_t)$ for stage t . These approximations are constructed using cutting planes derived by a separation approach and a reformulation derived by utilizing strong duality, respectively.

4.4.1 A Cutting Plane-Based Approximation for DRO-MSIP

We assume that the supports are finite (Assumption 4.4), and the valid cuts defined by (α_t^i, β_t^i) for $Q_{t+1}^{RA}(\cdot, \omega_{t+1}^i)$ for $t \in [T - 1]$ and $i \in \mathcal{N}_{t+1}$ are available. Given cuts (α_t^i, β_t^i) , for $i \in \mathcal{N}_{t+1}$, and solution \hat{x}_t , we identify a worst-case probability distribution by solving the following problem, referred to as *distribution separation problem* at stage $t \in [T - 1]$:

$$\max_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i \left((\alpha_t^i)^\top \hat{x}_t + \beta_t^i \right). \quad (4.21)$$

Let $\hat{P}_{t+1} = (\hat{p}_{t+1}^1, \dots, \hat{p}_{t+1}^{N_{t+1}})$ be an optimal solution to the distribution separation problem (4.21). We define a cutting plane-based approximating function for $t \in [T - 1]$ as

$$\phi_t^S(x_t) := \min \left\{ \phi : \phi \geq \pi_t^\top x_t + \gamma_t \right\}, \quad (4.22)$$

where $\pi_t = \sum_{i \in \mathcal{N}_{t+1}} \hat{p}_{t+1}^i \alpha_t^i$, and $\gamma_t = \sum_{i \in \mathcal{N}_{t+1}} \hat{p}_{t+1}^i \beta_t^i$. Clearly, the function ϕ_t^S under-approximates \mathcal{Q}_{t+1}^{RA} , since for solution $\bar{x}_t \neq \hat{x}_t$ it holds that $\mathcal{Q}_{t+1}^{RA}(\bar{x}_t) \geq \max_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i ((\alpha_t^i)^\top \bar{x}_t + \beta_t^i) \geq \phi_t^S(\bar{x}_t)$. Note that there are various types of ambiguity sets for which we can solve problem (4.21) and compute the coefficients (4.11) using a finite-time algorithm; e.g., ambiguity sets constructed using Wasserstein metric [36], moment information [106], total variation distance [10] and χ^2 distance [80], where their supports are finite.

4.4.2 A Reformulation-Based Approximation for DRO-MSIP

We present a reformulation-based approximation that relies on the dualization of the inner maximization problems in (4.3) and (4.4), i.e., $\max\{\mathbb{E}_{P_{t+1}}[Q_{t+1}^{RA}(\cdot, \cdot)] : P_{t+1} \in \mathcal{P}_{t+1}\}$ for $t \in [T - 1]$. We note that similar reformulation-based approaches are studied in recent papers for DRO-MSLPs with ϕ -divergence-based ambiguity sets [73], DRO-MSLPs with Wasserstein ambiguity sets [31], and DRO-MSIPs with moment-based ambiguity sets [106]. In the following, we demonstrate the DRO-SDDP-R algorithm for DRO-MSIPs with Wasserstein ambiguity sets, but we note that this approach can be applied to DRO-MSIPs with a general family of ambiguity sets for which dual formulations of DRO models are available.

The Wasserstein ambiguity set with finite support is defined for $t = 2, \dots, T$ as

$$\mathcal{P}_t = \left\{ P_t \in \mathbb{R}_+^{N_t} : \sum_{i \in \mathcal{N}_t} p_t^i = 1, \sum_{j \in \mathcal{N}_t} v_{ij} = p_t^i, i \in \mathcal{N}_t, \sum_{i \in \mathcal{N}_t} v_{ij} = \bar{p}_t^j, j \in \mathcal{N}_t, \right. \\ \left. \sum_{i \neq j \in \mathcal{N}_t} \|\boldsymbol{\xi}_t(\omega_t^i) - \boldsymbol{\xi}_t(\omega_t^j)\| v_{ij} \leq \epsilon_t, v_{ij} \geq 0, \forall i, j \in \mathcal{N}_t \right\},$$

where $\{\bar{p}_t^i\}_{i \in \mathcal{N}_t}$ is a reference probability distribution on Ω_t for $t \in \{2, \dots, T\}$. The dual of the maximization in (4.3) and (4.4) for $t \in [T - 1]$ is given as follows:

$$\min_{\rho_t \geq 0} \left\{ \epsilon_{t+1} \rho_t + \sum_{i \in \mathcal{N}_{t+1}} \bar{p}_{t+1}^i \nu_t^i : \right. \\ \left. \nu_t^i + \|\boldsymbol{\xi}_{t+1}(\omega_{t+1}^i) - \boldsymbol{\xi}_{t+1}(\omega_{t+1}^j)\| \rho_t \geq Q_{t+1}^{RA}(x_t, \omega_{t+1}^j), \forall i, j \in \mathcal{N}_{t+1} \right\}.$$

Here, the strong duality holds by Theorem 1 in Gao and Kleywegt [36]. Then, given cuts $(\alpha_t^i, \beta_t^i), i \in \mathcal{N}_{t+1}$, we define a reformulation-based approximating function for $t \in [T - 1]$ as

$$\phi_t^D(x_t) := \min \left\{ \epsilon_{t+1} \rho_t + \sum_{i \in \mathcal{N}_{t+1}} \bar{p}_{t+1}^i \nu_t^i : \rho_t \geq 0, \right. \quad (4.23a)$$

$$\left. \nu_t^i + \|\boldsymbol{\xi}_{t+1}(\omega_{t+1}^i) - \boldsymbol{\xi}_{t+1}(\omega_{t+1}^j)\| \rho_t \geq (\alpha_t^i)^\top x_t + \beta_t^i, \forall i, j \in \mathcal{N}_{t+1} \right\}. \quad (4.23b)$$

Remark 4.2. The convex approximation described in this section can be applied to DRO-MSIPs having continuous supports by employing additional algorithmic techniques. More specifically, the dual of the inner maximization in (4.3) and (4.4) with continuous support is given by the following semi-infinite program:

$$\min_{\rho_t \geq 0} \left\{ \epsilon_{t+1} \rho_t + \sum_{i \in \mathcal{N}_{t+1}} \bar{p}_{t+1}^i \nu_t^i : \right. \quad (4.24) \\ \left. \nu_t^i + \|\boldsymbol{\xi}_{t+1}(\omega_{t+1}^i) - \boldsymbol{\xi}_{t+1}(\omega_{t+1})\| \rho_t \geq Q_{t+1}^{RA}(x_t, \omega_{t+1}), \forall i \in \mathcal{N}_{t+1}, \omega_{t+1} \in \Omega_{t+1} \right\}.$$

Given $\omega_{t+1}^j \in \Omega_{t+1}$ and cuts (α_t^j, β_t^j) valid for $Q_{t+1}^{RA}(\cdot, \omega_{t+1}^j)$, for some j , we obtain valid cutting planes in the form of (4.23b) for this semi-infinite program. These constraints provide an outer approximation of the feasible region of the semi-infinite program. Consequently, the optimal value within the outer approximation is a lower bound on the value of (4.24). We iteratively improve the outer approximation through the following steps: (i) Solve the outer approximation of (4.24) with a finite number of cutting planes; (ii) Given its solution $(\bar{x}_t, \bar{\rho}_t)$, solve the separation problem: $\max\{Q_{t+1}^{RA}(\bar{x}_t, \omega_{t+1}) - \|\xi_{t+1}(\omega_{t+1}^i) - \xi_{t+1}(\omega_{t+1})\| \bar{\rho}_t : \omega_{t+1} \in \Omega_{t+1}\}$; (iii) Add a cutting plane in the form of (4.23b) associated with a scenario identified by solving the separation problem to the outer approximation of (4.24); (iv) Repeat until the outer approximation is as tight as a predetermined level. These iterative steps refine the outer approximation, making it increasingly accurate in representing the feasible region of the semi-infinite program, and thereby solving the DRO-MSIP with continuous support.

4.5 SDDP-Based Algorithms for DRR-MSP and DRO-MSP

4.5.1 Distributionally Ambiguous SDDP

We present a customized SDDP algorithm, referred to as *distributionally ambiguous SDDP* (DA-SDDP), for both DRO-MSIPs and DRR-MSIPs, where the convex approximations discussed in Sections 4.2 and 4.4 are utilized. The DA-SDDP approximates the pessimistic and optimistic expected cost-to-go functions Q_{t+1}^{RA} and Q_{t+1}^{RR} for each $t \in [T - 1]$. Notably, DA-SDDP shares the key sampling ideas of SDDP for approximating the expectation functions, but the crux of this algorithm is in deriving convex approximations and iterative refinement methods for them. We note that the conventional SDDP solves linear programs and uses dual solutions to derive a valid cut (or approximations) for MSLPs; however, this approach

Algorithm 4.1: Distributionally Ambiguous SDDP

```

Initialize:  $l \leftarrow 1$ ;  $x_0 \leftarrow$  initial state;  $\omega_1 \leftarrow$  data at the first stage;
                $\Omega_1 := \{\omega_1\}$ ;  $K_t^l \leftarrow 0$  for  $t = 1, \dots, T - 1$ ;
1 while (satisfying none of stopping conditions) do
2   Sample a scenario path  $\omega^{(l)} \in \Omega := \Omega_1 \times \dots \times \Omega_T$ ;
   // Forward Step
3   for  $t \in [T]$  do
4     Solve subproblem  $\mathbb{P}_t^l(x_{t-1}^l, \omega_t^{(l)})$  and obtain  $(x_t^l, y_t^l)$  and  $\hat{Q}_t^l(x_{t-1}^l, \omega_t^{(l)})$ ;
5   end
   // Backward Step
6   for  $t = T, \dots, 2$  do
7     for  $i \in \mathcal{N}_t$  do
8       Solve relaxation  $\tilde{\mathbb{P}}_t^l(x_{t-1}^l, \omega_t^i)$  and obtain cut  $(\alpha_{t-1}^{i,l}, \beta_{t-1}^{i,l})$ ;
9     end
10    Refine approximating function  $\phi_{t-1}^l$  by using cuts  $(\alpha_{t-1}^{i,l}, \beta_{t-1}^{i,l}), i \in \mathcal{N}_t$ ;
11     $K_{t-1}^l \leftarrow K_{t-1}^l + 1$ ;
12  end
13  Solve subproblem  $\mathbb{P}_1^l(x_0^l, \omega_1)$  and obtain the bound  $LB$   $K_t^{l+1} \leftarrow K_t^l$  for
      $t = 1, \dots, T - 1$ ;
14   $l \leftarrow l + 1$ ;
15 end
16 Return: Subproblems  $\{\mathbb{P}_t^l\}_{t \in [T-1]}$ ,  $LB$ 

```

is not applicable for DRR programs, requiring our convex approximations.

A pseudocode of DA-SDDP is given in Algorithm 4.1. 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 for $t \in [T - 1]$ at iteration l to 0. At iteration l , the algorithm samples a scenario path $\omega^{(l)} = (\omega_1^{(l)}, \dots, \omega_T^{(l)})$ from $\Omega := \Omega_1 \times \dots \times \Omega_T$ (Line 2). For the finite convergence of the algorithm, we assume that the scenario path sampling is conducted with replacement. 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 3 and 4) and a backward

step (Lines 6-13).

Forward Step. For each stage $t \in [T]$, DA-SDDP solves the following approximation of Problem (4.4) (or (4.6)), which we refer to as *subproblem* and denote by $\mathbb{P}_t^l(x_{t-1}^l, \omega_t^{(l)})$ (Line 4):

$$\hat{Q}_t^l(x_{t-1}^l, \omega_t^{(l)}) := \min_{(x_t, y_t) \in X_t(x_{t-1}^l, \omega_t^{(l)})} \left\{ f_t(x_t, y_t, \omega_t^{(l)}) + \phi_t^l(x_t) \right\}, \quad t \in [T], \quad (4.25)$$

where $\phi_T^l(\cdot) = 0$, and $x_{t-1}^l, t = 2, \dots, T$, is an optimal stage- $(t-1)$ solution. To simplify notation, we let $x_0^l := x_0$ and $X_1(x_0^l, \omega_1^{(l)}) := X_1$. Function $\phi_t^l(x_t)$, for each $t \in [T-1]$, is a 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— Q_{t+1}^{RA} and Q_{t+1}^{RR} —while solving DRO-MSIP and DRR-MSIP, respectively. The approximating functions presented in Section 4.2 and 4.4 are viable substitutes for the function ϕ_t^l when they are defined using the set of cuts $\{(\alpha_t^{i,k}, \beta_t^{i,k})\}_{k \in [K_t^l]}$ available at iteration l and stage t .

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

$$\hat{Q}_t^l(x_{t-1}, \omega_t^i) \geq (\alpha_{t-1}^{i,l})^\top x_{t-1} + \beta_{t-1}^{i,l}, \quad \forall x_{t-1} \in \{0, 1\}^{d_x}, i \in \mathcal{N}_t, \quad (4.26a)$$

$$\hat{Q}_t^l(x_{t-1}^l, \omega_t^i) = (\alpha_{t-1}^{i,l})^\top x_{t-1}^l + \beta_{t-1}^{i,l}. \quad (4.26b)$$

It should be noted that the cut's validity is defined for \hat{Q}_t^l , yet this condition is sufficient to construct a lower-approximation for the optimistic or pessimistic expected cost-to-go function, since \hat{Q}_t^l is always lower-bounding the exact value function Q_t^{RR} or Q_t^{RA} , 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 10). In Line 13, 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.

There are 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 (4.26a) and (4.26b). For example, by solving the subproblem to optimality, we can obtain an integer optimality cut, given a lower bound L for the value function \hat{Q}_t^l , in the following form:

$$\hat{Q}_t^l(x_{t-1}, \omega_t^i) \geq (\hat{q}_t^l - L) \left(\sum_{i \in [d_x]} 2x_{t-1,i}^l x_{t-1,i} - x_{t-1,i} - x_{t-1,i}^l \right) + \hat{q}_t^l,$$

where $\hat{q}_t^l := \hat{Q}_t^l(x_{t-1}^l, \omega_t^i)$. As another example, consider a Benders cut obtained by solving a linear programming relaxation of the subproblem when solving a DRO-MSIP (or DRR-MSIP). This cut satisfies (4.26a), though not necessarily (4.26b). 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 4.6.2) to solve the reformulation to optimality, and obtain a Benders cut that satisfies both (4.26a) and (4.26b).

4.5.2 Finite Convergence

Now, we show the finite convergence of DA-SDDP equipped with the convex approximations presented earlier. For DRR-MSIPs, we use DRR-SDDP-C and DRR-SDDP-R to denote variants of DA-SDDP with the cutting plane-based approximation ϕ_t^C (4.10) and the reformulation-based approximation ϕ_t^R (4.13), respectively. We define a *policy* by a collection of functions $\{\bar{x}_t(\omega_{[t]}), \bar{y}_t(\omega_{[t]})\}_{t \in [T]}$, where $\omega_{[t]} = (\omega_1, \dots, \omega_t)$, which serves as a decision rule given any scenario path $(\omega_1, \dots, \omega_T)$. A policy is *optimal* for a DRR-MSIP if $(\bar{x}_t(\omega_{[t]}), \bar{y}_t(\omega_{[t]}))$ is optimal to the t -th stage problem (4.6) ((4.5) for $t = 1$) for all stages $t \in [T]$ and all scenario paths $\omega_{[T]} \in \Omega$.

Theorem 4.4. *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 (4.11).*

Remark 4.3. When the ambiguity sets are singletons, the finite convergence result presented in Theorem 4.4 applies to MSIPs, where the finite convergence result previously established by Zou et al. [111]. While their proof utilizes assumptions regarding the validity, tightness, and finiteness of cuts, the proof in this chapter with binary state variables demonstrates that the convergence holds even without the finiteness assumption on cuts.

Theorem 4.5. *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.*

Similarly, for DRO-MSIPs, we refer to the variants of DA-SDDP as DRO-SDDP-C and DRO-

SDDP-R, utilizing the cutting plane-based approximation ϕ_t^S (4.22) and the reformulation-based approximation ϕ_t^D (4.23), respectively.

Theorem 4.6. *The forward step of the DRO-SDDP-C algorithm provides an optimal policy for DRO-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 (4.21).*

4.6 Extensions to Multistage Stochastic Disjunctive Programs with Distributional Ambiguity

In this section, we present extensions of the DA-SDDP algorithms to DRR- and DRO-MSDPs defined in Section 4.1.1, under the following assumption: For $h \in H_t$, set $\{(x_t, y_t) \in \mathbb{R}_+^{d_x} \times \mathbb{R}_+^{d_y} : A_t^h(\omega_t)x_t + B_t^h(\omega_t)y_t \geq b_t^h(\omega_t) - C_t^h(\omega_t)x_{t-1}\}$ is nonempty and compact for any $x_{t-1} \in \{0, 1\}^{d_x}$ and $\omega_t \in \Omega_t$. Also, its constraints include $x_t \geq 0$ and $x_t \leq 1$.

4.6.1 DA-SDDP Algorithms for DRR- and DRO-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 or optimistic expected cost-to-go function. Then, with the same definition of x_0 and ω_1 as in Algorithm 4.1, the subproblem at iteration l is given by

$$\hat{Q}_t^l(x_{t-1}, \omega_t) = \min_{(x_t, y_t) \in X_t(x_{t-1}, \omega_t)} \left\{ f_t(x_t, y_t, \omega_t) + \phi_t : \phi_t \geq (\pi_t^k)^\top x_t + \gamma_t^k, \quad k \in [K_t^l] \right\} \quad (4.27)$$

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 (4.7). We define the feasible set of the foregoing subproblem by

$$\begin{aligned} \mathcal{D}_t^l(x_{t-1}, \omega_t) := & \left\{ (x_t, y_t, \phi_t) \in \mathbb{R}_+^{d_x} \times \mathbb{R}_+^{d_y} \times \mathbb{R}_+ : \right. \\ & \bigvee_{h \in H_t} \left(\phi_t - (\pi_t^k)^\top x_t \geq \gamma_t^k, \quad k \in [K_t^l], \right. \\ & \left. \left. A_t^h(\omega_t)x_t + B_t^h(\omega_t)y_t \geq b_t^h(\omega_t) - C_t^h(\omega_t)x_{t-1} \right) \right\}, \quad (4.28) \end{aligned}$$

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

Proposition 4.7. *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 $\tilde{\mathcal{D}}_t^l(x_{t-1}, \omega_t)$ is given by*

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

Using Proposition 4.7, we also derive an extension of DA-SDDP for DRR- and DRO-MSDPs, namely DA-SDDP-DP. Its pseudocode is provided in Appendix B.3. 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 (4.27), derived using Proposition 4.7 and referred to as *LP-*

subproblems:

$$\min \left\{ f_t(x_t, y_t, \omega_t) + \phi_t : (x_t, y_t, \phi_t) \in \text{Proj}_{x_t, y_t, \phi_t} \left(\tilde{\mathcal{D}}_t^l(x_{t-1}, \omega_t) \right) \right\}, \quad (4.29)$$

for $t \in [T]$ and $\phi_T = 0$. We note that the DA-SDDP-DP algorithms for DRR- and DRO-MSDPs have the finite convergence if cuts $(\pi_{t-1}^l, \gamma_{t-1}^l)$ are obtained as in the cutting plane-based algorithms for DRR- and DRO-MSIPs, respectively. The comprehensive description of the algorithm is provided in Appendix B.3.

4.6.2 Application of Proposition 4.7 for Solving DRR- and DRO-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 DRO-MSIPs by applying Proposition 4.7. For the ease of exposition, let y_t be continuous. Then, the subproblem (4.25) of DRR- and DRO-MSIPs for iteration l and stage t can be rewritten as

$$\min \left\{ f_t(x_t, y_t, \omega_t^{(l)}) + \phi_t^l : \left(x_{t,j} = 0 \vee x_{t,j} = 1, j = 1, \dots, d_x \right) \right. \\ \left. \wedge \left(\phi_t - (\pi_t^k)^\top x_t \geq \gamma_t^k, k \in [K_t^l], \right. \right. \\ \left. \left. A_t(\omega_t^{(l)})x_t + B_t(\omega_t^{(l)})y_t \geq b_t(\omega_t^{(l)}) - C_t(\omega_t^{(l)})x_{t-1} \right) \right\}. \quad (4.30)$$

We use $\mathcal{D}_t^l(x_{t-1}, \omega_t^{(l)})$ and $\mathcal{D}_t^{l,LP}(x_{t-1}, \omega_t^{(l)})$ to denote the feasible region of the subproblem (4.30) and its linear programming relaxation, respectively. A relaxation of $\mathcal{D}_t^l(x_{t-1}, \omega_t^{(l)})$ can be defined as follows: $\mathcal{D}_t^{l,s}(x_{t-1}, \omega_t^{(l)}) := \mathcal{D}_t^{l,LP}(x_{t-1}, \omega_t^{(l)}) \cap \{x_{t,j} = 0 \vee x_{t,j} = 1, j \in [s]\}$, for $s \in [d_x]$. It is easy to see that the set $\mathcal{D}_t^{l,s}(\cdot, \cdot)$ for $s = d_x$ is equivalent to the

original set $\mathcal{D}_t^l(\cdot, \cdot)$. Moreover, $\mathcal{D}_t^{l,LP}(\cdot, \cdot) \supseteq \mathcal{D}_t^{l,1}(\cdot, \cdot) \supseteq \cdots \supseteq \mathcal{D}_t^{l,d_x}(\cdot, \cdot) = \mathcal{D}_t^l(\cdot, \cdot)$, and thus $\text{conv}(\mathcal{D}_t^{l,1}(\cdot, \cdot)) \supseteq \cdots \supseteq \text{conv}(\mathcal{D}_t^{l,d_x}(\cdot, \cdot)) = \text{conv}(\mathcal{D}_t^l(\cdot, \cdot))$. This provides a hierarchy of relaxations of the feasible region $\mathcal{D}_t^l(\cdot, \cdot)$ of DRR- and DRO-MSIPs. The tight extended formulation of the convex hull of the relaxations can be obtained using Proposition 4.7.

Proposition 4.8. *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 $\omega_t \in \Omega_t$:*

$$\left\{ \begin{array}{l} \sum_{h \in \llbracket \mathcal{J}_t^s \rrbracket} \zeta_{t,0}^h = 1, \quad \sum_{h \in \llbracket \mathcal{J}_t^s \rrbracket} \zeta_{t,1}^h - x_t = 0, \\ \sum_{h \in \llbracket \mathcal{J}_t^s \rrbracket} \zeta_{t,2}^h - y_t = 0, \quad \sum_{h \in \llbracket \mathcal{J}_t^s \rrbracket} \zeta_{t,3}^h = x_{t-1}, \quad \sum_{h \in \llbracket \mathcal{J}_t^s \rrbracket} \zeta_{t,4}^h - \phi_t = 0, \\ A_t(\omega_t)\zeta_{t,1}^h + B_t(\omega_t)\zeta_{t,2}^h + C_t(\omega_t)\zeta_{t,3}^h \geq b_t(\omega_t), \quad h \in \llbracket \mathcal{J}_t^s \rrbracket, \\ \zeta_{t,1,j}^h = 0, \quad j \in J_1^h, \quad h \in \llbracket \mathcal{J}_t^s \rrbracket, \\ \zeta_{t,1,j}^h = \zeta_{t,0}^h, \quad j \in J_2^h, \quad h \in \llbracket \mathcal{J}_t^s \rrbracket, \\ \zeta_{t,4}^h - (\pi_t^k)^\top \zeta_{t,1}^h - \gamma_t^k \zeta_{t,0}^h \geq 0, \quad h \in \llbracket \mathcal{J}_t^s \rrbracket, k \in [K_t^l], \\ x_t \in \mathbb{R}_+^{d_x}, y_t \in \mathbb{R}_+^{d_y}, \phi_t \in \mathbb{R}_+, \\ \zeta_{t,0}^h \in \mathbb{R}_+, \zeta_{t,1}^h \in \mathbb{R}_+^{d_x}, \zeta_{t,2}^h \in \mathbb{R}_+^{d_y}, \zeta_{t,3}^h \in \mathbb{R}_+^{d_x}, \zeta_{t,4}^h \in \mathbb{R}_+, \quad h \in \llbracket \mathcal{J}_t^s \rrbracket \end{array} \right\}, \quad (4.31)$$

where $\mathcal{J}_t^s := \{(J_1^h, J_2^h) : h \in \llbracket \mathcal{J}_t^s \rrbracket\}$, $s \in [d_x]$, $t \in [T]$, 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 $|J_1 \cup J_2| = s$.

Proposition 4.8 provides the following relaxation that can be used to generate cuts in Line 8 of Algorithm 4.1: $\tilde{Q}_t^{l,s}(x_{t-1}^l, \omega_t^{(l)}) = \min \left\{ f_t(x_t, y_t, \omega_t^{(l)}) + \phi_t^l(x_t) : (x_t, y_t) \in \text{Proj}_{x_t, y_t}(\mathcal{D}_t^{l,s}(x_{t-1}^l, \omega_t^{(l)})) \right\}$, where $\tilde{Q}_t^{l,1}(x_{t-1}^l, \omega_t^{(l)}) \leq \cdots \leq \tilde{Q}_t^{l,d_x}(x_{t-1}^l, \omega_t^{(l)}) = \hat{Q}_t^l(x_{t-1}^l, \omega_t^{(l)})$. For $s = d_x$, a Benders cut obtained by solving this relaxation is a supporting hyperplane satisfying (4.26b). 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. 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 4.4. The above relaxations can be readily extended to the case where y_t are mixed-binary variables. Furthermore, if y_t are mixed-integer, a hierarchy of relaxations can be derived by employing the binary expansion as discussed in Remark 4.1.

4.7 Computational Tests

In this section, we present computational results for the DRR-SDDP-C, DRR-SDDP-R, DRO-SDDP-C, and DRO-SDDP-R algorithms. Recall that these are specific implementations of DA-SDDP, utilizing the approximations (4.10), (4.13), (4.22), and (4.23), respectively. We apply these algorithms to solve instances of multistage maximum flow and facility location interdiction problems with distributional ambiguity. All the algorithms are implemented in Julia 1.8 where subproblems, coefficient-computing problems (4.11), and distribution separation problem (4.21) 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` [28] package to be consistent with the research community, e.g., [80], 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.

In our implementations, all the algorithms generate a strengthened Benders cut and an integer optimality cut alternately during the backward steps. For details on the strengthened Benders cuts, we refer readers to Zou et al. [111]. For ambiguity set, we consider Wasserstein ambiguity set with the l_1 norm throughout all test instances. Note that in this case the distribution separation problem (4.21) is a linear program and the subproblem (4.25) in the

DRR-SDDP-R algorithm is a mixed-binary linear program.

4.7.1 MS-MFIP with Distributional Ambiguity

Throughout this section, we consider DRR and DRO variants of the MS-MFIP formulation that is provided in Appendix B.1. In the subsequent sections, we present the results of a comparative analysis of the algorithms and demonstrate the significance of DRR and DRO for MS-MFIP. In addition, we extend the analysis to the case involving data corruption where the defender can deceive the interdictor by providing false information about critical arcs in a network.

4.7.1.1 Instance Generation and Computational Results

Networks are randomly generated, following the method presented in Cormican et al. [26]. 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 % 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 Ω_t of size $|\Omega|$ for each stage t . We set the Wasserstein ball size parameter ϵ to 30. The interdiction budget for each stage is set to one to avoid a trivial solution where

Table 4.1: Details of DRR- and DRO-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

an interdiction solution in an early stage completely separates the source and sink nodes, leaving no arcs to remove in later stages.

Table 4.1 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 i th 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 three hours for all algorithms. Also, there is an early-termination condition based on the convergence of lower bound. In particular, the algorithm is stopped if the lower bound fails to improve for 100 consecutive iterations.

In Table 4.2, we report lower bounds and solution times in seconds for each algorithm, labeled LBound and Time (s), respectively. We may also obtain upper bounds computed statistically

Table 4.2: Performance comparison of algorithms for DRR- and DRO-MFIP instances

Instance	DRO-SDDP-C		DRO-SDDP-R		DRR-SDDP-C		DRR-SDDP-R	
	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+

through a sampling approach using the policy after termination. However, to focus on the computational performance of the algorithms in terms of convergence while running, we only report lower bounds and solution times here. The results indicate that the DRO-SDDP-C algorithm provides better lower bounds and solution times than the DRO-SDDP-R algorithm for all instances. On average, the DRO-SDDP-C algorithm converges 17.2 times faster than the DRO-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 DRO-SDDP-R algorithm’s performance is more susceptible to T and $|\Omega|$ than the DRO-SDDP-C algorithm. For example, as $|\Omega|$ increases from 5 to 15 for NI-1-3-5, the DRO-SDDP-R algorithm’s solution time increases by 48.2 times, while the DRO-SDDP-C algorithm’s solution time increases by 2.7 times. Similarly, when T increases to 5 and 6 from 3 for NI-1-3-5, the DRO-SDDP-R algorithm fails to solve the 5-stage and 6-stage instances—

NI-1-5-5 and NI-1-6-5—within the time limit, whereas the DRO-SDDP-C algorithm solves all instances within the time limit. This is mainly because the DRO-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 converges, 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 (4.13b), incorporated by the linearization and the ambiguity set, respectively.

4.7.1.2 Impact of DRO and DRR on MS-MFIP

To demonstrate the impact of DRO and DRR on MS-MFIP, we present the results from out-of-sample tests, which are conducted as follows. We first sample realizations of the capacity over the stages. Then, the DRO-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 given a realization of the capacity. We simulate the DRR and DRO 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 in this section, 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 capacity realizations are sampled from a truncated normal distribution with a mean of 30, a standard deviation of 5, and values constrained to the interval $[10, 50]$. The policies are generated for a set of different values of the Wasserstein ball size parameter

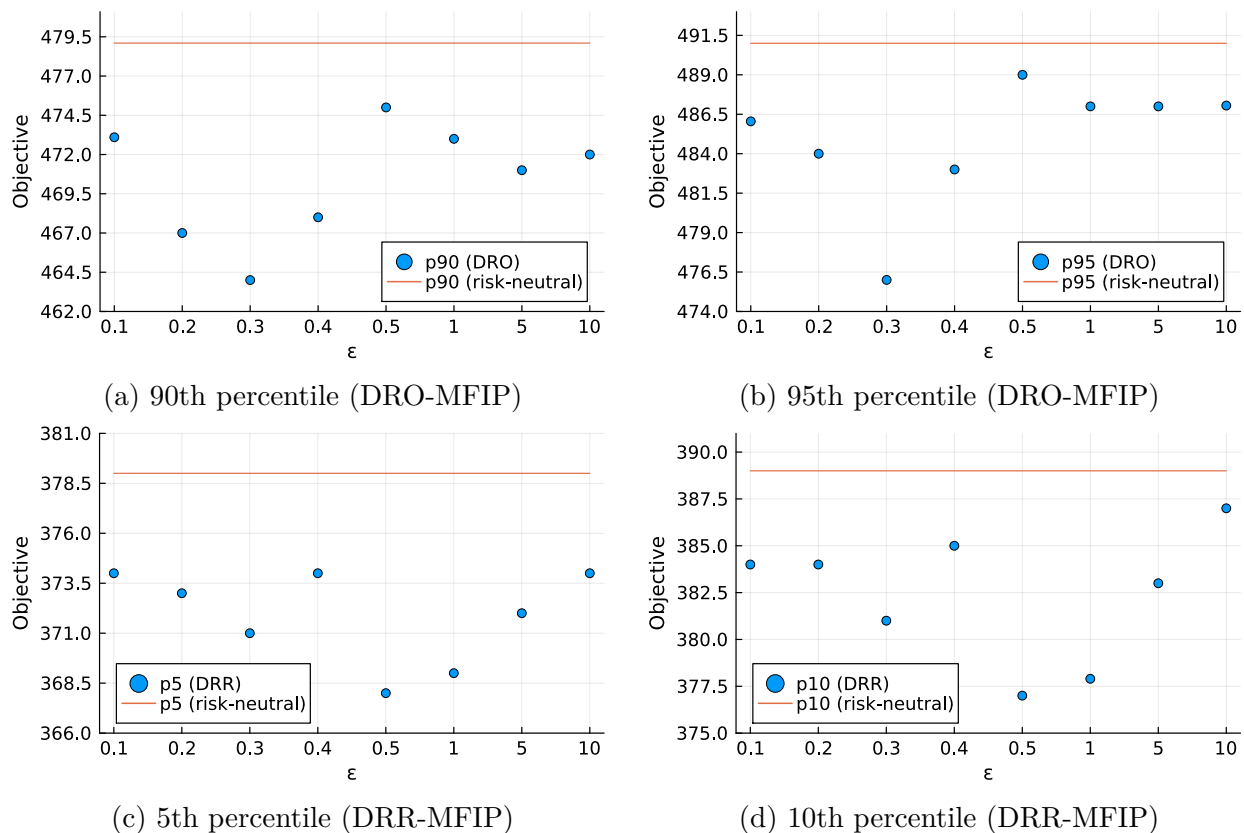


Figure 4.1: Plots of percentiles from out-of-sample tests on DRO-MFIP (90th and 95th percentiles) and DRR-MFIP (5th and 10th percentiles)

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 Figure 4.1: Figures 4.1a and 4.1b show the 90th and 95th percentiles of the objective values obtained by the DRO policies, and Figures 4.1c and 4.1d 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 DRO policies give the smaller 90th and 95th percentiles compared to the risk-neutral policy. As the ball size 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 the ball size 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 DRO policies achieve the robustness of interdiction solutions by incorporating conservatism over unfavorable scenarios 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 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.

4.7.1.3 Significance of DRR Framework for MS-MFIP with Data Corruption

In this test setup, we assume the network user has a capability of deceiving the interdictor by providing false information about critical arcs, i.e., the most essential arcs to maintain maximum flow. Specifically, the network user introduces uncertainty about the success of interdiction on the critical arcs before the interdictor makes a decision. In this adversarial setting, we consider a risk-receptive interdictor, i.e., an optimistic decision-maker, to mitigate the impact of the false information. The significance of DRR is demonstrated through out-of-sample tests we conduct as follows.

To simplify analysis, we focus on a two-stage case of MS-MFIP, where the interdiction decision is made only in the first stage, and the flow decision is made only in the second stage. That is, the interdictor's decision is made based on the corrupted sample data of the random vector in the second stage. Let $G = (N, A)$ represent the network. For random data generation, we generate samples of the capacity of each arc $a \in A$ from a normal distribution with mean μ_a and standard deviation $\sigma_a = \mu_a/4$, where μ_a is drawn from a discrete uniform

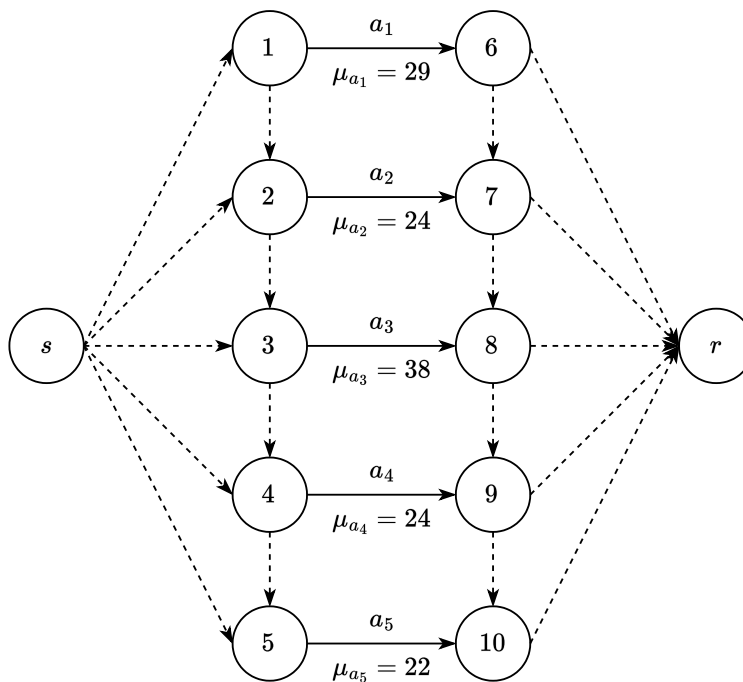


Figure 4.2: A network in a 5×2 grid shape.

distribution in the interval $[20, 40]$. The network user manipulates these sample data by solving deterministic MS-MFIP on G with true mean values, identifying the critical arcs in A , and introducing fake data where interdiction on these arcs fails, replacing an $\bar{\alpha}$ proportion of the sample data. The parameter $\bar{\alpha} \in [0, 1]$ denotes the contamination level. Throughout all tests, we fix the number of scenarios to 30 for consistency.

We first consider a simple network depicted in Figure 4.2. The solid lines represent interdiction arcs, and the dotted lines represent non-interdiction arcs. The label on each arc represents its name and mean of capacity. In this network, the most critical arc is a_3 , followed by a_1 as the second most critical. The third most critical arc is either a_2 or a_4 , as they share the same mean value. We assume that the network user manipulates the data associated with arcs a_1, a_2 , and a_3 . Now we assess the impact of the corrupted data and DRR on optimal interdiction solutions by solving the DRR variant of MS-MFIP with Wasserstein ambiguity set. We also solve the DRO variant for comparative purposes. The

Table 4.3: Optimal interdiction solutions for the DRR and DRO models with $\bar{\alpha} \in \{0.2, 0.4, 0.6\}$ and $\epsilon \in \{0, 10, 20, 40\}$.

$\bar{\alpha}$	ϵ						
	Risk-neutral	DRR			DRO		
	0	10	20	40	10	20	40
0.2	(a_3, a_4)	(a_1, a_3)	(a_1, a_3)	(a_2, a_3)	(a_3, a_4)	(a_4, a_5)	(a_4, a_5)
0.4	(a_3, a_4)	(a_1, a_3)	(a_1, a_3)	(a_2, a_3)	(a_4, a_5)	(a_4, a_5)	(a_4, a_5)
0.6	(a_4, a_5)	(a_4, a_5)	(a_1, a_3)	(a_1, a_3)	(a_4, a_5)	(a_4, a_5)	(a_4, a_5)

contamination level $\bar{\alpha}$ varies in $\{0.2, 0.4, 0.6\}$, and the Wasserstein ball size parameter ϵ varies in $\{0, 10, 20, 40\}$. The interdiction budget b is set to 2. In Table 4.3, we present the optimal interdiction solutions for these models. When the interdictor is risk-neutral, i.e., $\epsilon = 0$, solutions (a_3, a_4) and (a_4, a_5) are chosen. This demonstrates that the fake uncertainty of interdiction success motivates the interdictor to deviate from the optimal solution (a_1, a_3) with the true mean values by choosing arcs a_4 and a_5 , where the success of interdiction is guaranteed across all scenarios. The risk-averse interdictor prioritizes the most conservative solution (a_4, a_5) , illustrating how DRO could be susceptible to data corruption in adversarial settings. In contrast, the risk-receptive interdictor includes the critical arcs a_1, a_2 , and a_3 in their solutions in almost all cases. For $\bar{\alpha} = 0.6$, the optimal solution diverges from these critical arcs when $\epsilon = 10$, selecting (a_4, a_5) , but it reverts to interdicting the critical arcs as ϵ increases to 20 and 30. Next, we evaluate the out-of-sample performance of these interdiction solutions. We randomly sample 1000 scenario paths from the clean distributions and solve the second-stage problem using the solutions provided in Table 4.3 for $\bar{\alpha} = 0.4$. The plot of objective values for the sample paths is illustrated in Figure 4.3, with average values denoted by red marks along with their respective numbers. Since we are evaluating the interdictor's solutions, lower objective values indicate better performance. Comparing average performances, the DRR interdiction solutions, (a_1, a_3) and (a_2, a_3) , outperform the

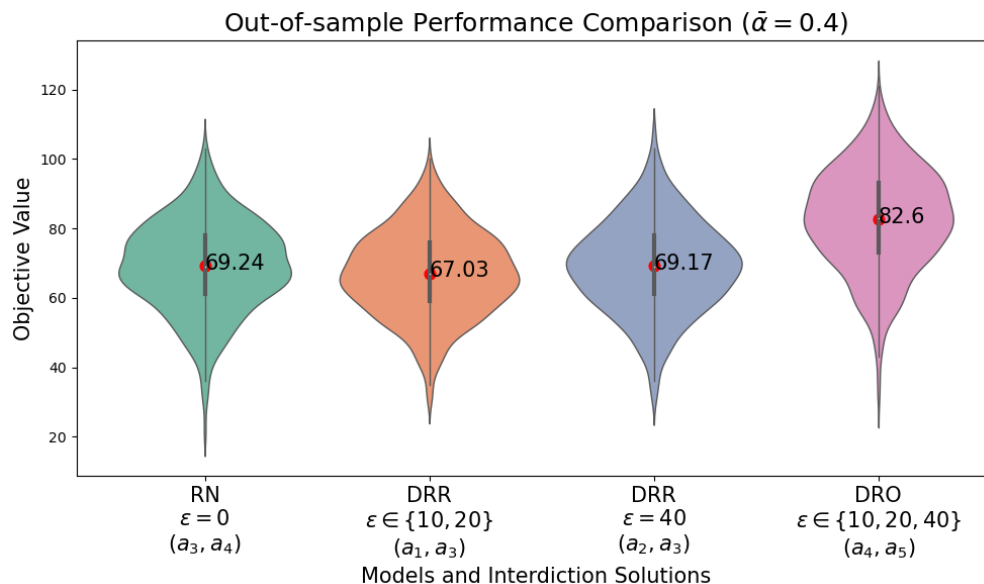


Figure 4.3: Comparison of interdiction solutions from different models in out-of-sample test. Network has a 5×2 grid shape, and contamination level $\bar{\alpha}$ is 0.4.

other solutions. The DRO interdiction solution (a_4, a_5) shows the most inferior out-of-sample performance among the four solutions with an average objective value that is 23.3 % higher than that of the DRR interdiction solution (a_1, a_3) .

To assess the significance of DRR in a more complicated instance, we perform additional tests with the network in a 6×3 grid shape and the interdiction budget of 5. We set $\bar{\alpha} = 0.4$, obtain the DRR and DRO solutions for $\epsilon \in \{0, 10, 20, 40\}$, and conduct the out-of-sample test with these solutions. The results are plotted in Figure 4.4. Consistent with our observations in the simpler instance, the out-of-sample performance of the DRR solutions improves as ϵ increases. In contrast, the DRO model produces identical solutions for all values of ϵ , yielding the poorest average performance among the four solutions.

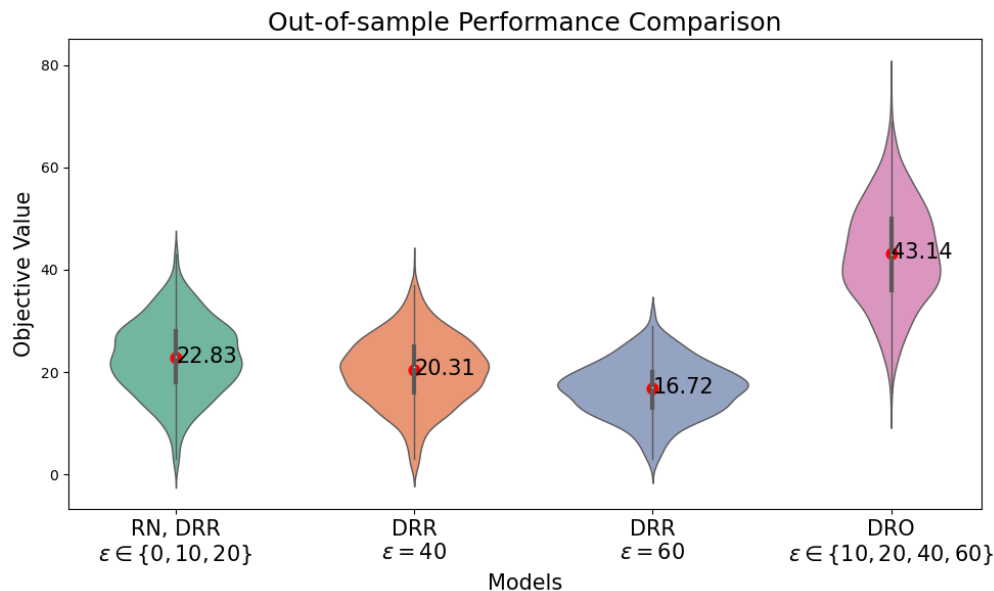


Figure 4.4: Comparison of interdiction solutions from different models in out-of-sample test. Network has a 6×3 grid shape, interdiction budget b is 5, and contamination level $\bar{\alpha}$ is 0.4.

4.7.2 MS-FLIP with Distributional Ambiguity

We evaluate the computational performance of the algorithms for MS-FLIP with distributional ambiguity, referred to as DRR-FLIP and DRO-FLIP. The detailed formulation of MS-FLIP is provided in Appendix B.1.

4.7.2.1 Instance Generation and Computational Results

To generate instances, we use the following method that is similar to the one used in [106]. 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 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

Table 4.4: Details of DRR- and DRO-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

[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 4.4. Each instance, denoted by LI- i - T - $|\Omega|$, involves the i th 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 DRO-MFIP instances.

Table 4.5 presents the upper bounds and solution times in seconds obtained by each algorithm 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 DRO-FLIP, the DRO-SDDP-C algorithm provides the upper bounds better than the DRO-SDDP-R algorithm for all instances. Also, on average, the DRO-SDDP-C

algorithm is 38.9 times faster than the DRO-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 DRO-SDDP-R algorithm is comparatively susceptible to the number of realizations per stage. For example, the DRO-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 DRO-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 DRO-SDDP-R and DRR-SDDP-R algorithms require more time to solve due to the larger subproblems resulting from the reformulation techniques.

4.8 Conclusion

We studied multistage stochastic integer and disjunctive programs under distributional ambiguity, considering the distributional risk-receptiveness and robustness in a decision making process. For distributionally risk-receptive multistage stochastic integer programs (DRR-MSIPs) without and with decision-dependent uncertainty, we presented new classes of cutting planes and reformulation-based approaches to derive convex approximations of the optimistic expected cost-to-go functions. For distributionally robust multistage stochastic integer programs (DRO-MSIPs), we presented cutting plane-based and reformulation-based approximations of the pessimistic expected cost-to-go functions. We developed algorithms using these approximations and provided their finite convergence analysis. Furthermore,

Table 4.5: Performance comparison of algorithms for DRR- and DRO-FLIP instances

Instance	DRO-SDDP-C		DRO-SDDP-R		DRR-SDDP-C		DRR-SDDP-R	
	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+

we extended the algorithms for distributionally risk-receptive and distributionally robust multistage stochastic disjunctive programs (DRR- and DRO-MSDPs) and then presented applications of them to solve DRR- and DRO-MSIPs using a hierarchy of relaxations. We compared the algorithms for DRR- and DRO-MSIPs by solving multistage stochastic network interdiction problems under distributional ambiguity that are multi-round attacker-defender games and have not been addressed in the literature. The computational results show that the algorithms using the cutting plane-based approximations outperform the algorithms using the reformulation-based approximations by 26.1 times, on average, in terms of solution time until convergence. In addition, we conducted out-of-sample tests, and their results demonstrate that the DRO 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. In an adversarial setting, we showcased that the DRR policies can be

used to mitigate the impact of data corruption.

Chapter 5.

Bi-Parameterized Two-Stage Stochastic Min-Max and Min-Min Mixed Integer Programs¹

5.1 Introduction

In this chapter, we investigate *bi-parameterized two-stage stochastic programming*, where the first-stage decisions affect both the constraints and the objective function in the recourse problem. For the readers' convenience, we restate the formulation of BTSPs as follows.

$$\min_{x \in X} \left\{ f(x) + \mathbb{E}[Q(x, \omega)] \right\}, \quad (5.1)$$

where vector x denotes the set of first-stage decision variables, Let $\boldsymbol{\xi} = (q, r, G, W, T)$ be a random vector associated with the second stage. For each scenario $\omega \in \Omega$, its realization occurs with probability $p(\omega)$. The recourse function $Q(x, \omega)$ for each scenario ω is defined as follows:

$$Q(x, \omega) := \min/\max \left\{ q(\omega)^\top y + x^\top G(\omega)y : W(\omega)y = r(\omega) - T(\omega)x, y \in \mathcal{Y} \right\}. \quad (5.2)$$

¹Reference [50]: Sumin Kang and Manish Bansal. Bi-parameterized two-stage stochastic min-max and min-min mixed integer programs. *Preprint arXiv:2501.01081*, 2025.

The notation “min/max” in (5.2) indicates that the recourse problem can either be a minimization problem or a maximization problem. Throughout this chapter, we refer to BTSP (5.1) as *min-min problem* when the recourse problem (5.2) is a minimization, and *min-max problem* when the recourse problem is a maximization problem. It can be easily seen that this formulation reduces to the conventional “single-parameterized” two-stage stochastic program when min/max replaced with min and $G(\omega) = 0$ for all $\omega \in \Omega$.

The first-stage feasible set is defined as $X := \{x \in \mathcal{X} : Ax = b\}$, where $\mathcal{X} \subseteq \mathbb{R}_+^{n_1}$ represents integrality restrictions on x , $A \in \mathbb{Q}^{m_1 \times n_1}$, and $b \in \mathbb{Q}^{m_1}$. The function $f : X \rightarrow \mathbb{R}$ is the first-stage objective function. Let $Y(x, \omega) := \{y \in \mathcal{Y} : W(\omega)y = r(\omega) - T(\omega)x\}$ for $\omega \in \Omega$ where $\mathcal{Y} \subseteq \mathbb{R}_+^{n_2}$ represents integrality restrictions on y . Here, $q(\omega) \in \mathbb{Q}^{n_2}$, $G(\omega) \in \mathbb{Q}^{n_1 \times n_2}$, $T(\omega) \in \mathbb{Q}^{m_2 \times n_1}$, $W(\omega) \in \mathbb{Q}^{m_2 \times n_2}$, and $r(\omega) \in \mathbb{Q}^{m_2}$.

A single-parameterized reformulation of BTSPs can be derived by introducing a proxy variable with constraints $\theta = x^\top G(\omega)y$. However, since the resulting constraints depend on the parameterized coefficient, i.e., $x^\top G(\omega)$ for y , the problem structure differs from the classical formulation of two-stage stochastic programming where coefficients associated with y are independent of x .

The bi-parameterization introduces significant computational challenges, mainly due to the nonconvexity of the recourse function, even without integrality restrictions on the second-stage variables y . To illustrate this, consider the following example:

Example 5.1. Let $\Omega = \{\frac{1}{3}, \frac{2}{3}, 1\}$ and $p(\omega) = 1/3$ for $\omega \in \Omega$. Consider the following recourse function: $Q(x, \omega) = \min\{(1+x)y_1 + (1+3x)y_2 : y_1 \geq x - \omega, y_2 \geq \omega - x, y \in \mathbb{R}_+^2\}$, for $\omega \in \Omega$. The expected recourse function, i.e., $\mathbb{E}[Q(x, \omega)] = \frac{1}{3}Q(x, \frac{1}{3}) + \frac{1}{3}Q(x, \frac{2}{3}) + \frac{1}{3}Q(x, 1)$, on $[0, 1]$ is nonconvex and nonsmooth, as illustrated in Figure 5.1. ■

Furthermore, when integrality restrictions are imposed on y , the recourse function Q becomes

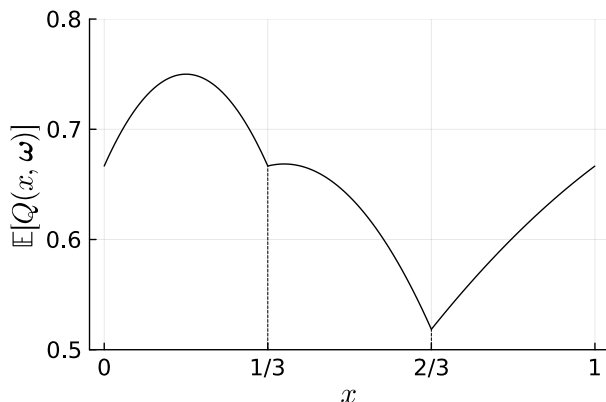


Figure 5.1: An example of a nonconvex expected recourse function.

discontinuous.

These properties of the recourse function limit the applicability of traditional decomposition methods. For example, consider the Benders decomposition (or the L-shaped method). When applied, it is unclear how to generate optimality cuts that are valid for the nonconvex recourse function $Q(x, \omega)$. In the case of single-parameterized recourse, a common approach is to solve the LP dual of the continuous relaxation of the recourse problem and use the dual solutions to construct optimality cuts—these cuts, while not necessarily tight, are valid. In contrast, for BTSPs, such cuts are not valid for the recourse function, making classical Benders decomposition methods inapplicable. Alternatively, the dual decomposition method [22] can be applied to BTSPs, but it results in computationally expensive subproblems. In this approach, copies of x , denoted by $x(\omega)$, and the nonanticipativity constraints $x = x(\omega)$ for $\omega \in \Omega$ are introduced and then relaxed using Lagrangian multipliers. This allows for scenario-wise decomposition; however, the resulting subproblem in each scenario is a mixed-integer nonconvex program, which can be computationally demanding to solve.

5.1.1 Applications of Bi-Parameterized Stochastic Programs

Despite the challenges involved in solving BTSPs, they have important applications to various classes of problems. We provide examples below and discuss these in more detail in Section 5.6.

- **Bi-parameterized network interdiction models.** Network interdiction problems involve a sequential game between two players: an interdictor and a network user, with conflicting objectives. These problems have diverse applications in practice, such as disrupting illicit supply networks [62, 64], planning military operation [86], and analyzing vulnerabilities in critical infrastructure [20]. The formulation of stochastic network interdiction problems is given by the following min-max form:

$$\min_{x \in X} \mathbb{E} \left[\max_{y \in Y(x, \omega)} f(x, y, \omega) \right],$$

where x denotes interdictor's decision variables and y represents network user's decision variables. The function $f(x, y, \omega)$ and set $Y(x, \omega)$ for $\omega \in \Omega$ are the network user's objective function and feasible set, respectively, and both of them depend on x . The formulation of bi-parameterized interdiction models has been considered a general form of interdiction problems in the literature [93]; however, to the best of our knowledge, no standard decomposition approaches have been established to address this general setup.

- **Stochastic optimization with decision-dependent uncertainty.** Stochastic optimization often involves decision-dependent uncertainty, where a decision affects the distribution (for example, see [30, 41]). Decision-dependent two-stage (risk-neutral) stochas-

tic programs can be formulated as:

$$\min_{x \in X} \left\{ f(x) + \mathbb{E}_{p(x)} [Q^s(x, \omega)] \right\},$$

where $Q^s(x, \omega) = \min\{q(\omega)^\top y : W(\omega)y = r(\omega) - T(\omega)x, y \in \mathcal{Y}\}$ is the recourse function of the classical single-parameterized problem, for each $\omega \in \Omega$. Here, the expectation is taken with respect to the decision-dependent distribution $p(x) = (p(x, \omega))_{\omega \in \Omega}^\top$, which we denote by $\mathbb{E}_{p(x)}$. By incorporating $p(x, \omega)$ into the objective function of each recourse problem $Q^s(x, \omega)$, resulting in $(p(x, \omega)q(\omega)^\top y)$, the above formulation can be easily reformulated into the min-min problem (5.1).

Furthermore, we can also consider a risk measure to model risk-averse behaviors of the decision-maker with the decision-dependent probabilities. For instance, decision-dependent two-stage risk-averse stochastic programs with conditional value-at-risk (CVaR) can be formulated as:

$$\min_{x \in X} \left\{ f(x) + \text{CVaR}_\alpha(Q^s(x, \omega), p(x)) \right\},$$

where the CVaR is given by (Rockafellar and Uryasev [83]):

$$\text{CVaR}_\alpha(Q^s(x, \omega), p(x)) = \min_{\eta \in \mathbb{R}} \eta + \frac{1}{1 - \alpha} \mathbb{E}_{p(x)} \left[(Q^s(x, \omega) - \eta)_+ \right], \quad (5.3)$$

which is a special case of the min-min problem (5.1).

- **Distributionally robust optimization with decision-dependent ambiguity.** Distributionally robust optimization (DRO) is an optimization framework that addresses uncertain parameters whose true distribution is unknown. In a DRO model, we consider a set of potential distributions—referred to as *ambiguity set*—and make decisions that

hedge against worst-case distributions within this set. The ambiguity set may depend on the first-stage decision x ; for example, see [7, 51, 60, 106]. A DRO model with decision-dependent ambiguity set can be formulated as:

$$\min_{x \in X} \left\{ f(x) + \max_{p \in \mathcal{P}(x)} \mathbb{E}_p [Q^s(x, \omega)] \right\},$$

where $\mathcal{P}(x)$ is a decision-dependent ambiguity set and $Q^s(x, \omega)$ is defined as above for $\omega \in \Omega$. When the support Ω is finite, the ambiguity set $\mathcal{P}(x)$ is often defined by a polyhedron (e.g., Wasserstein ambiguity set, moment-matching ambiguity set, and ϕ -divergence ambiguity set). In this case, by taking dual of the continuous relaxations of recourse (minimization) problems, we can derive a min-max (BTSP) reformulation of the DRO problem, which is a special case of the min-max problem (5.1).

Non-Relatively Complete Ambiguity Set. A well-known method for this decision-dependent DRO problem is to dualize the inner maximization and reformulate the entire problem as a single minimization problem. This dual-based method, however, requires the ambiguity set $\mathcal{P}(x)$ to be nonempty for all $x \in X$, i.e., “relatively complete ambiguity set”. This assumption can be impractical, particularly when the ambiguity set is constructed to match the moment information of the new distribution with the empirical distribution of sample data; e.g., refer to Basciftci et al. [7] and Yu and Shen [106]. In Section 5.7.3, we demonstrate through numerical results that the dual-based approach fails to solve instances where $\mathcal{P}(x)$ is empty for some $x \in X$. We also show the method presented in this chapter for BTSPs effectively addresses this issue, providing an exact solution to DRO problems without assuming the nonemptiness of $\mathcal{P}(x)$ for all $x \in X$.

5.1.2 Contributions and Organization

In this chapter, we introduce BTSPs, where first-stage decisions affect both the objective function and constraints in the recourse problem. These models have integrality restrictions in both the first and second stages. To efficiently solve these problems, we propose Lagrangian-integrated L-shaped (L^2) methods. Specifically, we present two exact algorithms for the min-max and min-min cases when x is binary, and a regularization-augmented algorithm for the mixed-binary case. We also introduce extensions of the L^2 method to address DRO variants of BTSPs (DR-BTSPs) with finite or continuous support of random parameters. The numerical results demonstrate the efficiency of the L^2 methods. It solves all tested instances of network interdiction problems in 23 seconds on average, while the benchmark method fails to solve any instances within the time limit of 1 hour. Additionally, the L^2 method for min-min programs outperforms existing methods, achieving optimal solutions, on average, 18.4 and 1.7 times faster for risk-neutral and DR bi-parameterized stochastic facility location problems, respectively.

Furthermore, we demonstrate that our approach can effectively address DRO problems with decision-dependent ambiguity sets that are not necessarily relatively complete, i.e., $\mathcal{P}(x)$ may be empty for some $x \in X$; we note that this setting remains unaddressed in the literature (e.g., [7, 60, 106]). For these problem instances, the L^2 method achieves optimal solutions, on average, 5.3 times faster than current approaches.

The remainder of the chapter is organized as follows. In Section 5.2, we introduce an exact decomposition method for the min-max problem, and in Section 5.3, we present its extension to solve the min-min problem. In Section 5.4, we propose a regularization-augmented variant of our decomposition method. We introduce DR-BTSPs and algorithms to solve them in Section 5.5, explore the applications of BTSPs in detail in Section 5.6, and present

computational results in Section 5.7. In Section 5.8, we make concluding remarks.

We make the following assumption on the feasible sets throughout this chapter:

Assumption 5.1. The set X is nonempty and compact. Also, for all $x \in X$, the sets $Y(x, \omega)$ for $\omega \in \Omega$ are nonempty and compact.

5.2 An Exact Algorithm for the Min-Max BTSP

In this section, we introduce a decomposition method referred to as the Lagrangian-integrated L-shaped (L^2) method to exactly solve the min-max BTSP (5.1) with integer variables in both stages.

5.2.1 A Lagrangian Dual Formulation

We begin by deriving a Lagrangian dual of the min-max problem. To do so, we first rewrite the recourse problem (5.2) in the following form by introducing proxy variables z for x :

$$Q(x, \omega) = \max q(\omega)^\top y + x^\top G(\omega)y \quad (5.4a)$$

$$\text{s.t. } T(\omega)z + W(\omega)y = r(\omega), \quad (5.4b)$$

$$z = x, \quad (5.4c)$$

$$z \in X, y \in \mathcal{Y}. \quad (5.4d)$$

It is important to note that we substitute x with z only in the constraints, while leaving x unchanged in the objective function—ensuring that the problem remains linear for fixed x . We then relax constraints (5.4c) with Lagrangian multipliers $\lambda(\omega) \in \mathbb{R}^{n_1}$. Then, the

Lagrangian relaxation is given by $(L(x, \lambda(\omega), \omega) - \lambda(\omega)^\top x)$ where

$$L(x, \lambda(\omega), \omega) = \max q(\omega)^\top y + x^\top G(\omega)y + \lambda(\omega)^\top z \quad (5.5a)$$

$$\text{s.t. } T(\omega)z + W(\omega)y = r(\omega), z \in X, y \in \mathcal{Y}. \quad (5.5b)$$

For any $\lambda(\omega) \in \mathbb{R}^{n_1}$, the Lagrangian relaxation $(L(x, \lambda(\omega), \omega) - \lambda(\omega)^\top x)$ provides an upper bound on $Q(x, \omega)$. Therefore, the tightest upper bound is obtained by solving the following Lagrangian dual:

$$D(x, \omega) := \min_{\lambda(\omega) \in \mathbb{R}^{n_1}} \left\{ L(x, \lambda(\omega), \omega) - \lambda(\omega)^\top x \right\}. \quad (5.6)$$

Assumption 5.2. In the min-max problem (5.1), variables x are mixed-integer. Let $I \subseteq [n_1]$ denote the index set corresponding to the components of x that affect the recourse function $Q(x, \omega)$. We assume that for all $i \in I$, x_i is binary, i.e., $x_i \in \{0, 1\}$.

Under the above assumption, we show that strong duality holds for the Lagrangian dual. It is worth noting that, although similar duality results have been established in the literature (e.g., Zou et al. [111]), our result is distinct in that we consider bi-parameterized problems and relax only the constraint parameterization, while leaving x in the objective function.

Theorem 5.1. *Under Assumption 5.2, $Q(x, \omega) = D(x, \omega)$ for any $\omega \in \Omega$ and $x \in X$, i.e., strong duality holds for the Lagrangian dual (5.6).*

Consequently, the min-max problem (5.1) can be exactly reformulated as

$$\min \left\{ f(x) + \sum_{\omega \in \Omega} p(\omega) \left(L(x, \lambda(\omega), \omega) - \lambda(\omega)^\top x \right) : x \in X, \lambda(\omega) \in \mathbb{R}^{n_1}, \forall \omega \in \Omega \right\}. \quad (5.7)$$

An advantage of addressing the reformulated problem instead of the original problem is that the Lagrangian function $L(\cdot, \cdot, \omega)$ for each $\omega \in \Omega$ is jointly convex on $\text{conv}(X) \times \mathbb{R}^{n_1}$. This

convexity enables the use of a subgradient method, which decomposes the computation of L . However, nonconvex terms $\lambda(\omega)^\top x$ appear in the objective function, which can cause computational challenges.

To address this issue, we introduce an analytical form for the optimal Lagrangian multipliers.

Lemma 5.2. *There exist real values $\sigma_i \in \mathbb{R}_+$ for $i \in I$ such that, for any $\hat{x} \in X$ and $\omega \in \Omega$, the solution $\hat{\lambda}(\omega) = (\hat{\lambda}_1(\omega), \dots, \hat{\lambda}_{n_1}(\omega))$, defined by*

$$\hat{\lambda}_i(\omega) = \begin{cases} -\sigma_i & \text{if } i \in I \text{ and } \hat{x}_i = 0 \\ \sigma_i & \text{if } i \in I \text{ and } \hat{x}_i = 1, \\ 0 & \text{otherwise} \end{cases} \quad (5.8)$$

is optimal to the Lagrangian dual (5.6). Here, $I \subseteq [n_1]$ is the index set corresponding to the components of x that affect the recourse function $Q(x, \omega)$.

Furthermore, the following provides a sufficient condition on the values of σ to guarantee the above optimality:

$$\sum_{i \in I: \hat{z}_i(\omega) \neq \hat{x}_i} \sigma_i \geq \max_{y \in Y(\hat{x}, \omega)} \left\{ (q(\omega) + G(\omega)^\top \hat{x})^\top (\hat{y}(\omega) - y) \right\}, \quad (5.9)$$

where $(\hat{y}(\omega), \hat{z}(\omega))$ is an optimal solution of the Lagrangian relaxation (5.5) given $(\hat{x}, \hat{\lambda}(\omega), \omega)$.

Using this analytical form for an optimal solution, we can derive an exact reformulation of the min-max problem as follows.

Theorem 5.3. *Let $(\sigma_i)_{i \in I}$ satisfy the condition (5.9) for all $\hat{x} \in X$ and $\omega \in \Omega$. The following*

problem is an exact reformulation of the min-max problem (5.1):

$$\min_{x \in X} \left\{ f(x) - \sum_{i \in I} \sigma_i x_i + \sum_{\omega \in \Omega} p(\omega) \hat{Q}(x, \omega) \right\} \quad (5.10)$$

where, for each $\omega \in \Omega$,

$$\begin{aligned} \hat{Q}(x, \omega) &= \max q(\omega)^\top y + x^\top G(\omega) y + \sum_{i \in I} \sigma_i (2x_i - 1) z_i \\ &\text{s.t. } T(\omega) z + W(\omega) y = r(\omega), \quad z \in X, \quad y \in \mathcal{Y}. \end{aligned} \quad (5.11)$$

Remark 5.1. To find values $\sigma_i, i \in I$, that satisfy condition (5.9) for all $\hat{x} \in X$ and $\omega \in \Omega$, we can use the following upper and lower bounds: $\bar{U}B \geq \max_{y \in Y(x, \omega)} (q(\omega) + G(\omega)^\top x)^\top y$ and $\bar{L}B \leq \min_{y \in Y(x, \omega)} (q(\omega) + G(\omega)^\top x)^\top y$, respectively. Notice that their difference is always greater than or equal to the right-hand side of the condition (5.9), that is, $\max_{y \in Y(\hat{x}, \omega)} (q(\omega) + G(\omega)^\top \hat{x})^\top (\hat{y}(\omega) - y) = (q(\omega) + G(\omega)^\top \hat{x})^\top \hat{y}(\omega) - \min_{y \in Y(\hat{x}, \omega)} (q(\omega) + G(\omega)^\top \hat{x})^\top y \leq \bar{U}B - \bar{L}B$. By setting σ_i equal to this difference ($\bar{U}B - \bar{L}B$) for $i \in I$, we can ensure that the condition (5.9) is satisfied.

Alternatively, we may use an optimization-based approach to compute these values. For example, define $\Delta_y(x, \omega) = \{y^1 - y^2 : y^1, y^2 \in Y(x, \omega)\}$ and consider $\sigma_i(\omega) = \max \left\{ \left(\frac{q(\omega)}{|I|} + G_i(\omega) x_i \right)^\top \delta_y : x \in X, \delta_y \in \Delta_y(x, \omega) \right\}$ for $i \in I$ and $\omega \in \Omega$, where $G_i(\omega)$ denotes the i th column of $G(\omega)$. Notice that $\sum_{i \in I} \sigma_i(\omega) \geq \max \{ (q(\omega) + G(\omega)^\top x)^\top \delta_y : x \in X, \delta_y \in \Delta_y(x, \omega) \}$ for $\omega \in \Omega$. This inequality holds since, for an optimal solution (x^*, δ_y^*) to the right-hand side problem, the solution (x_i^*, δ_y^*) is feasible for the corresponding i th problem defining $\sigma_i(\omega)$. These $\sigma_i(\omega)$ values ensure that the condition (5.9) is satisfied for each $\omega \in \Omega$. By taking the maximum over $\omega \in \Omega$, i.e., $\sigma_i = \max_{\omega \in \Omega} \sigma_i(\omega)$, we have σ_i values that satisfy the condition (5.9) for all $\hat{x} \in X$ and $\omega \in \Omega$. ■

Algorithm 5.1: The L^2 method for the min-max problem

```

1 Initialization:  $l \leftarrow 1, LB \leftarrow -\infty, UB \leftarrow \infty, x^1 \in X$ ;
2 while  $(UB - LB)/UB > \epsilon^{tol}$  do
3   Solve the subproblems, obtain  $\{(y^l(\omega), z^l(\omega))\}_{\omega \in \Omega}$ , and compute  $\hat{Q}(x^l, \omega)$ ;
4   Compute  $\phi^l = f(x^l) - \sum_{i \in I} \sigma_i^\top x_i^l + \sum_{\omega \in \Omega} p(\omega) \hat{Q}(x^l, \omega)$ ;
5   if  $\phi^l < UB$  then
6      $UB \leftarrow \phi^l; x^* \leftarrow x^l$ ;
7   end
8   Add an optimality cut to master problem;
9   Solve the master problem and obtain optimal solution  $(x^{l+1}, \hat{\theta})$ ;
10   $LB \leftarrow f(x^{l+1}) - \sum_{i \in I} \sigma_i^\top x_i^{l+1} + \hat{\theta}$ ;
11   $l \leftarrow l + 1$ ;
12 end
13 Return:  $x^*$ ;

```

5.2.2 The Lagrangian-Integrated L-Shaped Method

We now present the L^2 method for solving the min-max problem (5.1) through its reformulation (5.10). At each iteration, the L^2 method adds a valid cut, referred to as *optimality cut*, that approximates the function $\sum_{\omega \in \Omega} p(\omega) \hat{Q}(x, \omega)$ in (5.10). The pseudo-code of the L^2 method is outlined in Algorithm 5.1. It starts by initializing iteration counter l to 1, lower bound LB to $-\infty$, and upper bound UB to ∞ . Let $x^1 \in X$ be any feasible solution. For each iteration l , we first solve the subproblems for $\omega \in \Omega$, which is given by the formulation (5.11) with $x = x^l$. Subsequently, we compute the objective value at the current solution x^l , denoted by ϕ^l . If this value ϕ^l is less than the current upper bound UB , then it replaces UB , and the current solution is saved as the best-known solution. Next, using the optimal solutions to the subproblems, an optimality cut is obtained and then added to the

master problem. In iteration l , master problem is given as follows:

$$\min \left\{ f(x) - \sum_{i \in I} \sigma_i x_i + \theta : x \in X, \right. \\ \left. \theta \geq \sum_{\omega \in \Omega} p(\omega) \left(q(\omega)^\top y^k(\omega) + x^\top G(\omega) y^k(\omega) + \sum_{i \in [n_1]} \sigma_i (2x_i - 1) z_i^k(\omega) \right), \forall k \in [l] \right\}, \quad (5.12)$$

where $(y^k(\omega), z^k(\omega))_{k \in [l]}$ are the optimal solutions of the subproblems at the corresponding iterations. In the following step, we solve the master problem, obtain a first-stage solution x^{l+1} to be explored in the next iteration, and replace the lower bound LB with the objective value of the master problem. We repeat this procedure until the optimality gap is less than or equal to a predetermined tolerance level ϵ^{tol} .

Proposition 5.4. *Let values $\sigma_i \in \mathbb{R}_+, i \in I$, satisfy the condition (5.9) for all $x \in X$ and $\omega \in \Omega$, and let $\epsilon^{tol} = 0$. Under Assumption 5.2, Algorithm 5.1 terminates in a finite number of iterations with x^* optimal to the min-max BTSP (5.1).*

Remark 5.2. To accelerate Algorithm 5.1, we can fix z to x^l when solving the subproblem in Algorithm 5.1. Solving this restricted subproblem with $z = x^l$ yields a feasible solution to the original problem (5.11), allowing us to derive an optimality cut valid for $\hat{Q}(x, \omega)$. This approach reduces computational effort while still providing a valid cut.

5.3 An Exact Algorithm for the Min-Min BTSP

The L^2 method for the min-min problem (5.1) begins by deriving its min-max reformulation. To this end, we convexify the recourse feasible region $Y(x, \omega)$, which is achieved sequentially by adding parametric inequalities. We make the following assumption for the exactness of our algorithm.

Assumption 5.3. In the min-min problem, all first-stage variables x are binary, i.e., $\mathcal{X} = \{0, 1\}^{n_1}$.

Let $S(\omega) := \{(x, y) \in X \times \mathcal{Y} : T(\omega)x + W(\omega)y = r(\omega)\}$ be a lifted feasible set in the (x, y) -space for $\omega \in \Omega$. Assume that $\tilde{T}(\omega) \in \mathbb{Q}^{m(\omega) \times n_1}$, $\tilde{W}(\omega) \in \mathbb{Q}^{m(\omega) \times n_2}$, and $\tilde{r}(\omega) \in \mathbb{Q}^{m(\omega)}$ appropriately sized matrices and vectors such that $\text{conv}(S(\omega)) = \{(x, y) \in \mathbb{R}_+^{n_1} \times \mathbb{R}_+^{n_2} : \tilde{T}(\omega)x + \tilde{W}(\omega)y \geq \tilde{r}(\omega)\}$. The intersection of $\text{conv}(S(\omega))$ and the hyperplane $\{(x, y) : x = \hat{x}\}$ for any $\hat{x} \in X$ is a face of $S(\omega)$, as x is binary, thereby all extreme points in the intersection have the y components in \mathcal{Y} . By this observation, for any $\hat{x} \in X$, we have

$$Q(\hat{x}, \omega) = \min \left\{ q(\omega)^\top y + \hat{x}^\top G(\omega)y : (x, y) \in \text{conv}(S(\omega)) \cap \{(x, y) : x = \hat{x}\} \right\} \quad (5.13)$$

$$= \min \left\{ q(\omega)^\top y + \hat{x}^\top G(\omega)y : \tilde{W}(\omega)y \geq \tilde{r}(\omega) - \tilde{T}(\omega)\hat{x}, y \in \mathbb{R}_+^{n_2} \right\}. \quad (5.14)$$

Without loss of generality, we can assume $m(\omega)$ is finite since $Y(x, \omega)$ is bounded. That is, the convex hull $\text{conv}(S(\omega))$ can be represented with finitely many parametric inequalities in the form of $(\alpha^1)^\top x + (\alpha^2)^\top y \geq \beta$; e.g., lift-and-project cuts [3] and Gomory cuts [38]. Thus, the linear programming dual of (5.14) is given by

$$Q(x, \omega) = \max \left\{ \pi^\top (\tilde{r}(\omega) - \tilde{T}(\omega)x) : \tilde{W}(\omega)^\top \pi \leq q(\omega) + G(\omega)^\top x, \pi \in \mathbb{R}_+^{m(\omega)} \right\}. \quad (5.15)$$

Then, we can rewrite the min-min problem as the following min-max problem:

$$\min_{x \in X} \left\{ f(x) + \sum_{\omega \in \Omega} p(\omega) \max \left\{ \pi^\top (\tilde{r}(\omega) - \tilde{T}(\omega)x) : \tilde{W}(\omega)^\top \pi \leq q(\omega) + G(\omega)^\top x, \pi \in \mathbb{R}_+^{m(\omega)} \right\} \right\}. \quad (5.16)$$

By applying the result in Theorem 5.3 to the above min-max reformulation, we obtain the fol-

lowing another reformulation of the min-min problem: $\min_{x \in X} f(x) - \sigma^\top x + \sum_{\omega \in \Omega} p(\omega) \hat{Q}(x, \omega)$ where

$$\hat{Q}(x, \omega) = \max \pi^\top (\tilde{r}(\omega) - \tilde{T}(\omega)x) + \sum_{i \in [n_1]} \sigma_i (2x_i - 1) z_i \quad (5.17a)$$

$$\text{s.t. } \tilde{W}(\omega)^\top \pi - G(\omega)^\top z \leq q(\omega) \quad (5.17b)$$

$$z \in X, \pi \in \mathbb{R}_+^{m(\omega)}. \quad (5.17c)$$

Now we present the L^2 method for the min-min problem. Its pseudo-code is outlined in Algorithm 5.2. The algorithm sequentially convexifies the set $S(\omega)$, i.e., it constructs the information $\{\tilde{T}(\omega), \tilde{W}(\omega), \tilde{r}(\omega)\}_{\omega \in \Omega}$ by adding parametric inequalities to subproblems. To this end, we consider an oracle that provides a violated parametric inequality if there exists any, which is the cut-generating linear program presented in Balas et al. [3]. We refer to such an oracle as the *cut-generating* oracle.

Algorithm 5.2 starts by initializing iteration counter l to 1, lower bound LB to $-\infty$, and upper bound UB to ∞ . Let $x^1 \in X$ be any feasible solution. At each iteration l , we first solve the following problems, called *primal subproblems*:

$$\min \left\{ q(\omega)^\top y + (x^l)^\top G(\omega)y : W^{l-1}(\omega)y \geq r^{l-1}(\omega) - T^{l-1}(\omega)x^l, y \in \mathbb{R}_+^{n_2} \right\}, \quad (5.18)$$

where $(T^0(\omega), W^0(\omega), r^0(\omega)) = (T(\omega), W(\omega), r(\omega))$, for $\omega \in \Omega$. The matrices and vector are updated to $(T^l(\omega), W^l(\omega), r^l(\omega))$ through the following procedure. Let $y^l(\omega)$ be the optimal solution to the primal subproblems for $\omega \in \Omega$. If $y^l(\omega) \notin \mathcal{Y}$, then the cut-generating oracle is used, and a violated parametric inequality is added to the primal subproblem. Otherwise, we keep the current matrices and vector. Notice that the primal subproblem is a relaxation of the recourse problem (5.14), thereby through this approach the L^2 method refines the

Algorithm 5.2: The L^2 method for the min-min problem

```

1 Initialization:  $l \leftarrow 1, LB \leftarrow -\infty, UB \leftarrow \infty, x^1 \in X$ ;
2 while  $(UB - LB)/UB > \epsilon^{tol}$  do
3   for  $\omega \in \Omega$  do
4     Solve the primal subproblem and obtain optimal solution  $y^l(\omega)$ ;
5     if  $y^l(\omega) \notin \mathcal{Y}$  then
6       Find a violated parametric inequality using the cut-generating oracle;
7        $(T^l(\omega), W^l(\omega), r^l(\omega)) \leftarrow (T^{l-1}(\omega), W^{l-1}(\omega), r^{l-1}(\omega))$  appended with the
          information of the parametric inequality;
8     else
9        $(T^l(\omega), W^l(\omega), r^l(\omega)) \leftarrow (T^{l-1}(\omega), W^{l-1}(\omega), r^{l-1}(\omega))$ ;
10    end
11  end
12  Solve the dual subproblems, obtain  $\{(\pi^l(\omega), z^l(\omega))\}_{\omega \in \Omega}$ , and compute  $\hat{Q}^l(x^l, \omega)$ ;
13  Compute  $\psi^l = f(x^l) - \sigma^\top x^l + \sum_{\omega \in \Omega} p(\omega) \hat{Q}^l(x^l, \omega)$ ;
14  if  $y^l(\omega) \in \mathcal{Y}, \forall \omega \in \Omega$  and  $\psi^l < UB$  then
15     $UB \leftarrow \psi^l; x^* \leftarrow x^l$ ;
16  end
17  Add an optimality cut to master problem;
18  Solve the master problem and obtain optimal solution  $(x^{l+1}, \hat{\theta})$ ;
19   $LB \leftarrow f(x^{l+1}) - \sigma^\top x^{l+1} + \hat{\theta}$ ;
20   $l \leftarrow l + 1$ ;
21 end
22 Return:  $x^*$ ;

```

relaxation iteratively. In the next step, we solve the following problems for $\omega \in \Omega$, referred to as *dual subproblems*:

$$\hat{Q}^l(x^l, \omega) = \max \left\{ \pi^\top (r^l(\omega) - T^l(\omega)x^l) + \sum_{i \in [n_1]} \sigma_i (2x_i^l - 1) z_i : \right. \\ \left. W^l(\omega)^\top \pi - G(\omega)^\top z \leq q(\omega), z \in X, \pi \in \mathbb{R}_+^{m^l} \right\}. \quad (5.19)$$

where σ is a given vector that satisfies the condition (5.9) for all $\omega \in \Omega$ and $x \in X$. Note that the dimension m^l of variables π varies over iterations corresponding to the updates in the information $(T^l(\omega), W^l(\omega), r^l(\omega))$. After solving the dual subproblems, an under-

approximation ψ^l of the objective value at the current solution x^l is computed; this objective value is exact if $y^l(\omega) \in \mathcal{Y}$ for all $\omega \in \Omega$. If ψ^l is exact and $\psi^l < UB$, then we update the upper bound, and the current solution is marked as the best solution so far. Subsequently, using the optimal solutions of the dual subproblems, an optimality cut is computed and added to *master* problem that is given by

$$\min \left\{ f(x) - \sigma^\top x + \theta : x \in X, \right. \quad (5.20a)$$

$$\left. \theta \geq \sum_{\omega \in \Omega} p(\omega) \left(\pi^k(\omega)^\top (r^k(\omega) - T^k(\omega)x) + \sum_{i \in [n_1]} \sigma_i z_i^k(\omega) (2x_i - 1) \right), \forall k \in [l] \right\}. \quad (5.20b)$$

Constraints (5.20b) are optimality cuts added for iteration $k \in [l]$ where $(\pi^k(\omega), z^k(\omega))_{k \in [l]}$ are optimal solutions of the dual subproblems at the corresponding iterations.

Proposition 5.5. *The optimality cut obtained at iteration l is valid for $\sum_{\omega \in \Omega} p(\omega) \hat{Q}(x, \omega)$, i.e., $\sum_{\omega \in \Omega} p(\omega) \hat{Q}(x, \omega) \geq \sum_{\omega \in \Omega} p(\omega) \left(\pi^l(\omega)^\top (r^l(\omega) - T^l(\omega)x) + \sum_{i \in [n_1]} \sigma_i z_i^l(\omega) (2x_i - 1) \right)$.*

Next, we solve the master problem and obtain a new lower bound on the optimal objective value. It also identifies a solution x^{l+1} to be explored in the next iteration. This process is repeated until the optimality gap, $(UB - LB)/UB$, equals or falls below a predetermined tolerance level $\epsilon^{tol} \geq 0$.

5.4 A Regularization-Augmented Algorithm for BTSPs

The proposed exact methods (Algorithms 5.1 and 5.2) require a predetermined value of σ that satisfies the condition (5.9). In this section, we introduce an alternative approach that does not require specifying σ a priori. Instead of relying on the analytical form in Lemma 5.2, we directly address the bilinear problem (5.7). Specifically, we propose a subgradient method

and improve its computational efficiency by stabilizing updates of the Lagrangian multipliers through reformulation and regularization techniques. This method provides an approximate solution for BTSPs with general mixed-binary x , that is, for the cases where Assumptions 5.2 and 5.3 do not hold.

For simplicity, in this section, we focus on the min-max problem where first-stage variables x are general mixed-binary and the functions $L(x, \lambda(\omega), \omega)$ for $\omega \in \Omega$ are defined as (5.5). We note that the proposed method can be adapted to the min-min problem in a similar manner after taking the min-max reformulation, described in Section 5.3.

We begin by discussing how stabilizing multiplier updates can improve computational efficiency. Consider a standard subgradient method (see, e.g., Bertsekas [15]) that iteratively updates both x and $\lambda(\omega)$. An issue arises in updating $\lambda(\omega)$, depending on the values of x . For example, suppose x is pure binary and fixed at \hat{x} , and that the subgradient method has converged to the optimal multipliers $\lambda_i(\omega) = \sigma_i(2\hat{x}_i - 1)$ for $i \in [n_1]$, where σ satisfies condition (5.9). Now, consider $i \in [n_1]$ such that $\hat{x}_i = 0$, so $\lambda_i(\omega) = -\sigma_i$. If, in the next iteration, \hat{x}_i is switched to 1, the optimal multiplier changes to σ_i . This requires a change of $2\sigma_i$ in the multiplier, which may demand many update steps and lead to slow convergence.

Motivated by this observation, we propose a reformulation that prevents sign flips in updating targets. The key idea is to set the multipliers to be nonpositive when $x_i = 0$ and nonnegative when $x_i = 1$ for the binary variables and to update their magnitudes at each iteration, rather than updating the multipliers directly. Let $N' \subseteq [n_1]$ be the index set of binary variables in

Algorithm 5.3: The regularized L^2 method for the min-max problem

```

1 Initialization:  $l \leftarrow 1, LB \leftarrow -\infty, UB \leftarrow \infty, x^1 \in X, \mu^1(\omega) \in \mathbb{R}_+^{n_1}$  for  $\omega \in \Omega$ ;
2 while  $(UB - LB)/UB > \epsilon^{tol}$  do
3   Solve the subproblems (5.22), obtain  $\{(y^l(\omega), z^l(\omega))\}_{\omega \in \Omega}$ , and compute
    $\hat{Q}^l(x^l, \mu^l(\omega), \omega)$ ;
4   Compute  $\phi^l = f(x^l) - \sum_{\omega \in \Omega} p(\omega) \mu^l(\omega)^\top x^l + \sum_{\omega \in \Omega} p(\omega) \hat{Q}^l(x^l, \mu^l(\omega), \omega)$ ;
5   if  $\phi^l < UB$  then
6      $UB \leftarrow \phi^l; x^* \leftarrow x^l$ ;
7   end
8   Add an optimality cut to master problem (5.23);
9   Solve master problem (5.23) and obtain  $x^{l+1}, \{\mu^{l+1}(\omega)\}_{\omega \in \Omega}, \{\eta^{l+1}(\omega)\}_{\omega \in \Omega}$ , and  $\hat{\theta}$ ;
10   $LB \leftarrow f(x^{l+1}) + \hat{\theta} - \sum_{\omega \in \Omega} p(\omega) \sum_{i \in [n_1]} \eta_i^{l+1}(\omega) + \gamma R(\mu^{l+1})$ ;
11   $l \leftarrow l + 1$ ;
12 end
13 Return:  $x^*$ ;
```

x , and let $N'' = [n_1] \setminus N'$. The reformulation is given by

$$\min f(x) + \sum_{\omega \in \Omega} p(\omega) \left(L(x, \lambda(\omega), \omega) - \lambda(\omega)^\top x \right) \quad (5.21a)$$

$$\text{s.t. } x \in X \quad (5.21b)$$

$$\lambda_i(\omega) = \mu_i(\omega)(2x_i - 1), \quad \forall \omega \in \Omega, i \in N' \quad (5.21c)$$

$$\mu_i(\omega) \in \mathbb{R}_+, \quad \forall \omega \in \Omega, i \in N' \quad (5.21d)$$

$$\lambda_i(\omega) \in \mathbb{R}, \quad \forall \omega \in \Omega, i \in N''. \quad (5.21e)$$

Here, a variable $\mu_i(\omega)$ represents the magnitude of the corresponding multiplier $\lambda_i(\omega)$, for each $i \in N'$. Thus, updating this $\mu_i(\omega)$ for $i \in N'$ instead of $\lambda_i(\omega)$ in a subgradient method can stabilize the update process. When x is pure binary, the reformulation (5.21) is exact, as an optimal solution, given in Lemma 5.2, is feasible to this problem. If x is mixed-binary, then it provides an approximate solution to the original problem.

We now present a variant of the L^2 method tailored to the above reformulation. We re-

fer to this algorithm as the *regularized L^2 method*, as it incorporates a regularization term into the master problem. This regularization term is introduced to make the optimal multiplier unique and prevent having excessively large multiplier values. The pseudo-code of the regularized L^2 method is outlined in Algorithm 5.3. We describe its details below for completeness. It starts by initializing iteration counter l to 1, bounds LB to $-\infty$, and UB to ∞ . Initial feasible solutions are denoted by x^1 and $\lambda^1(\omega)$ for $\omega \in \Omega$. Next, we solve the subproblems, which are defined as follows, given x^l and $\lambda^l(\omega)$ for $\omega \in \Omega$:

$$\hat{Q}^l(x^l, \lambda^l(\omega), \omega) = \max \left\{ q(\omega)^\top y + (x^l)^\top G(\omega)y + \lambda^l(\omega)^\top z : \right. \\ \left. T(\omega)z + W(\omega)y = r(\omega), z \in X, y \in \mathcal{Y} \right\}. \quad (5.22)$$

Let $(y^l(\omega), z^l(\omega))$ be the optimal solution of the subproblem for $\omega \in \Omega$. Using the optimal objective values of the subproblems, we can obtain an under-approximation ϕ^l of the optimal objective value of the min-max problem. This under-approximation can then be used to update the upper bound UB . In the following step, we add an optimality cut to the master problem given as follows:

$$\min_{x \in X} f(x) + \theta - \sum_{\omega \in \Omega} p(\omega) \lambda(\omega)^\top x + \gamma R(\mu) \quad (5.23a)$$

$$\text{s.t. (5.21b)–(5.21e)} \quad (5.23b)$$

$$\theta \geq \sum_{\omega \in \Omega} p(\omega) \left(q(\omega)^\top y^k(\omega) + x^\top G(\omega)y^k(\omega) + \lambda(\omega)^\top z^k(\omega) \right), \quad \forall k \in [l], \quad (5.23c)$$

where $\mu = (\mu(\omega))_{\omega \in \Omega}$. The term $R(\mu)$ denotes a regularization term, such as ℓ_1 regularization $\|\mu\|_1$ or Tikhonov regularization $\|\mu\|_2^2$, with $\|\cdot\|_p$ denoting the ℓ^p norm. Here, γ is a predetermined parameter that controls the significance of the regularization term. Let x^{l+1} , $(\lambda^{l+1}(\omega))_{\omega \in \Omega}$, $\mu^{l+1} = (\mu^{l+1}(\omega))_{\omega \in \Omega}$, and $\hat{\theta}$ be an optimal solution of the master problem at iteration l . In Algorithm 5.3, the lower bound LB is updated using the optimal objective

value of the master problem.

5.5 Distributionally Robust BTSPs with Finite and Continuous Support

In this section, we extend the L^2 method for solving distributionally robust BTSP (DR-BTSP), which is formulated as follows:

$$\min_{x \in X} \left\{ f(x) + \max_{p \in \mathcal{P}} \mathbb{E}_P[Q(x, \omega)] \right\} \quad (5.24)$$

where $Q(x, \omega) = \min/\max\{q(\omega)^\top y + x^\top G(\omega)y : W(\omega)y = r(\omega) - T(\omega)y, y \in \mathcal{Y}\}$ for $\omega \in \Omega$, and \mathcal{P} is an ambiguity set. We present the extended approaches for both finite-support and continuous-support cases of DR-BTSP. For the ease of exposition, we only present results for maximization recourse problem which can then be utilized to derive solution approaches for minimization problem as well.

For finite support, we address the following reformulation of DR-BTSPs, derived as in Theorem 5.3, using the Lagrangian dual and the analytical form of the Lagrangian multipliers.

Proposition 5.6. *Let vector $\sigma = (\sigma_1, \dots, \sigma_{n_1}) \in \mathbb{R}_+^{n_1}$ satisfies the condition (5.9) for all $\omega \in \Omega$ and $\hat{x} \in X$. The following problem is an exact reformulation of DR-BTSP (5.24):*

$$\min_{x \in X} \left\{ f(x) - \sigma^\top x + \max_{p \in \mathcal{P}} \sum_{\omega \in \Omega} p(\omega) \hat{Q}(x, \omega) \right\} \quad (5.25)$$

where, for each $\omega \in \Omega$, $\hat{Q}(x, \omega)$ is given by (5.11).

For any feasible $\hat{p} \in \mathcal{P}$, we have $\max_{p \in \mathcal{P}} \sum_{\omega \in \Omega} p(\omega) \hat{Q}(x, \omega) \geq \sum_{\omega \in \Omega} \hat{p}(\omega) \hat{Q}(x, \omega)$. Therefore,

we can derive an optimality cut for $\max_{p \in \mathcal{P}} \sum_{\omega \in \Omega} p(\omega) \hat{Q}(x, \omega)$ by aggregating optimality cuts for $\hat{Q}(x, \omega)$, for $\omega \in \Omega$, with respect to $\hat{p} \in \mathcal{P}$. The strongest cut can be obtained by determining the worst-case distribution p through an optimal solution of the following *distribution separation problem*:

$$\max_{p \in \mathcal{P}} \sum_{\omega \in \Omega} p(\omega) \hat{Q}(x, \omega). \quad (5.26)$$

Note that the distribution separation problem becomes a linear program in many cases, including those with ambiguity sets defined by moment information, ϕ -divergence, or Wasserstein metric. Building on this observation, an extended L^2 method for DR-BTSPs is given by Algorithm 5.1 with the following modifications:

- (1) After solving the subproblems, add an additional step to solve the distribution separation problem for a fixed $x = x^l$. Let p^l denote an optimal solution to the distribution separation problem.
- (2) Compute $\phi^l = f(x^l) - \sigma^\top x^l + \sum_{\omega \in \Omega} p^l(\omega) \hat{Q}^l(x^l, \omega)$.
- (3) Generate an optimality cut of the form:

$$\theta \geq \sum_{\omega \in \Omega} p^l(\omega) \left(q(\omega)^\top y^l(\omega) + x^\top G(\omega) y^l(\omega) + \sum_{i \in [n_1]} \sigma_i (2x_i - 1) z_i^l(\omega) \right),$$

where $(y^l(\omega), z^l(\omega))$ is an optimal solution to subproblem (5.11) with $x = x^l$ for $\omega \in \Omega$ at iteration l .

For continuous support, we present our results for Wasserstein ambiguity set, defined using Wasserstein metric as follows:

$$\mathcal{P} := \left\{ p \in \mathcal{M}(\Omega) : \mathcal{W}(p, \bar{p}) \leq \epsilon^w \right\}. \quad (5.27)$$

Here, $\mathcal{M}(\Omega)$ is a set of all probability distributions defined on Ω , \bar{p} is a reference distribution, e.g., empirical distribution in a data-driven setting, and $\mathcal{W}(p, p')$ is the Wasserstein distance between distributions p and p' , which is defined as follows:

$$\mathcal{W}(p, p') := \inf_{\pi \in \mathcal{M}(\Omega \times \Omega)} \left\{ \mathbb{E}_{\pi} \left[\|\boldsymbol{\xi}(\omega) - \boldsymbol{\xi}(\omega')\| \right] : \pi^{\omega} = p, \pi^{\omega'} = p' \right\}, \quad (5.28)$$

where $\mathcal{M}(\Omega_t \times \Omega_t)$ is the set of all joint probability distributions supported on $\Omega_t \times \Omega_t$, π^{ω} denote the marginal distribution of ω , and $\|\cdot\|$ represents an arbitrary norm. The ambiguity set is interpreted as a ball which contains all probability distributions within a predetermined radius ϵ^w from the reference distribution \bar{p} .

Consider a data-driven setting where we have a finite sample $\bar{\Omega} = \{\omega^1, \dots, \omega^S\}$. Let \bar{p} be the empirical distribution on this sample, i.e., $\bar{p}(\omega^s) = \frac{1}{S} \delta_{\omega^s}$ where δ_{ω} is the Dirac delta function centered at $\omega \in \Omega$. Using the strong duality result [36] for the inner maximization, we can reformulate DR-BTSP (5.24) as the following problem:

$$\min f(x) + \epsilon^w \alpha + \frac{1}{S} \sum_{s \in [S]} \beta_s \quad (5.29a)$$

$$\text{s.t. } x \in X, \alpha \geq 0 \quad (5.29b)$$

$$\beta_s \geq Q(x, \omega) - \alpha \|\boldsymbol{\xi}(\omega) - \boldsymbol{\xi}(\omega^s)\|, \forall \omega \in \Omega, s \in [S]. \quad (5.29c)$$

Solving the above problem involves addressing the infinitely many constraints (5.29c). However, this can be achieved through a cut-generating approach, where violated constraints are identified and added, while solving the problem, using the following separation problem: $\max_{\omega \in \Omega} \{Q(x, \omega) - \alpha \|\boldsymbol{\xi}(\omega) - \boldsymbol{\xi}(\omega^s)\|\}$ for $s \in [S]$ given (x, α) . Although this separation problem is nonconvex, it can be reformulated into a mixed-integer program with some additional conditions on the uncertainty (e.g., see Duque and Morton [31]).

Additionally, solving the dual form (5.29) requires deriving cuts that approximate the rhs of constraints (5.29c), which can be achieved using our approaches. By Theorem 5.1, constraint (5.29c) for each $\omega \in \Omega$ is equivalent to $\min_{\lambda} L(x, \lambda, \omega) - \lambda^\top x - \alpha \|\xi(\omega) - \xi(\omega^s)\|$ where the Lagrangian function $L(x, \lambda, \omega)$ is given by (5.5). We assume that the uncertainty only affects the objective coefficients $q(\omega)$ of the recourse problem; therefore, $G(\omega) = G, T(\omega) = T, W(\omega) = W$, and $r(\omega) = r$ with appropriately sized data G, T, W , and r . For fixed ω , applying the analytical form of the multipliers, with σ satisfying the condition (5.9), we have valid cutting planes of the form:

$$\beta_s \geq -\sigma^\top x + q(\omega)^\top \bar{y} + x^\top G\bar{y} + \sum_{i \in [n_1]} \sigma_i (2x_i - 1) \bar{z}_i - \alpha \|\xi(\omega) - \xi(\omega^s)\|, \quad (5.30)$$

where (\bar{y}, \bar{z}) is a solution in $\{(y, z) \in \mathcal{Y} \times X : Tz + Wy = r\}$. By using these cutting planes within the cut-generating approach, we can derive a decomposition algorithm that utilizes the separation problem, the subproblem (5.11), and the following master problem:

$$\min f(x) + \epsilon^w \alpha + \frac{1}{S} \sum_{s \in S} \beta_s \quad (5.31a)$$

$$\text{s.t. } x \in X, \alpha \geq 0 \quad (5.31b)$$

$$(5.30), \forall \omega \in \Omega^l, s \in [S], \quad (5.31c)$$

where the set $\Omega^l \subseteq \Omega$ is expanded iteratively by adding scenario ω^l , obtained from solving the separation problem.

5.6 BTSP-Based Reformulations for Stochastic Optimization and Interdiction Problems

In this section, we present the details of BTSP-based reformulations for decision-dependent (risk-averse) stochastic optimization and generalized interdiction problems, introduced in Section 5.1. For clarity, we restate the single-parameterized recourse function Q^s , which will be used throughout this section, as follows:

$$Q^s(x, \omega) = \min\{q(\omega)^\top y : W(\omega)y = r(\omega) - T(\omega)x, y \in \mathcal{Y}\}.$$

5.6.1 Risk-Averse Stochastic Optimization with Decision-Dependent Probability Distribution

We consider the following formulation of two-stage stochastic programs with decision-dependent probabilities and CVaR measure:

$$\min_{x \in X} \left\{ f(x) + \text{CVaR}_\alpha \left(Q^s(x, \omega), p(x) \right) \right\}. \quad (5.32)$$

Using the linear programming formulation of CVaR (5.3), we can rewrite this problem as follows:

$$\min f(x) + \eta + \frac{1}{1 - \alpha} \sum_{\omega \in \Omega} p(x, \omega) \nu(\omega) \quad (5.33a)$$

$$\text{s.t. } \nu(\omega) \geq Q^s(x, \omega) - \eta, \quad \forall \omega \in \Omega \quad (5.33b)$$

$$x \in X, \eta \in \mathbb{R}, \nu(\omega) \in \mathbb{R}_+, \forall \omega \in \Omega. \quad (5.33c)$$

Suppose that $p(\cdot, \omega)$ for each scenario $\omega \in \Omega$ is an affine function. The above formulation can be addressed using a dual decomposition-based approach, as described in [90]. However, this approach only yields a lower bound on the optimal objective value, thereby leading to a duality gap potentially. Additionally, as discussed in Section 5.1, applying the dual decomposition method to this formulation results in mixed-integer nonconvex subproblems, which can impose a significant computational burden.

Now, we present a reformulation of (5.32) into the form of the min-max BTSP (5.1). By taking the dual of (5.3), we can obtain the dual representation of $\text{CVaR}_\alpha(Q^s(x, \omega), p(x))$:

$$\max \left\{ \frac{1}{1-\alpha} \sum_{\omega \in \Omega} \gamma(\omega) Q^s(x, \omega) : \sum_{\omega \in \Omega} \gamma(\omega) = 1 - \alpha, 0 \leq \gamma(\omega) \leq p(x, \omega), \forall \omega \in \Omega \right\}. \quad (5.34)$$

Let $Q^s(x, \omega) = \max\{\pi(\omega)^\top (r(\omega) - T(\omega)x) : W(\omega)^\top \pi(\omega) \leq q(\omega), \pi(\omega) \in \mathbb{R}_+^{m_2}\}$ be the dual formulation of the recourse problem with a convexified feasible region for each scenario ω . By incorporating this into the dual representation of CVaR, we have

$$\text{CVaR}_\alpha(Q^s(x, \omega), p(x)) = \max \frac{1}{1-\alpha} \sum_{\omega \in \Omega} \gamma(\omega) \pi(\omega)^\top (r(\omega) - T(\omega)x) \quad (5.35a)$$

$$\text{s.t. } \sum_{\omega \in \Omega} \gamma(\omega) = 1 - \alpha \quad (5.35b)$$

$$\gamma(\omega) \leq p(x, \omega), \quad \forall \omega \in \Omega \quad (5.35c)$$

$$W(\omega)^\top \pi(\omega) \leq q(\omega), \quad \forall \omega \in \Omega \quad (5.35d)$$

$$\pi(\omega) \in \mathbb{R}_+^{m_2}, \gamma(\omega) \in \mathbb{R}_+, \quad \forall \omega \in \Omega. \quad (5.35e)$$

For any $\gamma(\omega) \geq 0$, the inequality $(W(\omega)^\top \pi(\omega) \leq q(\omega))$ holds if and only if $(W(\omega)^\top \gamma(\omega) \pi(\omega) \leq q(\omega) \gamma(\omega))$. Therefore, we can replace (5.35d) with $(W(\omega)^\top \gamma(\omega) \pi(\omega) \leq q(\omega) \gamma(\omega))$ for each $\omega \in \Omega$. Next, we introduce a decision vector $\tau(\omega) \in \mathbb{R}_+^{m_2}$ to substitute for $\gamma(\omega) \pi(\omega)$, as for any $\tau(\omega) \geq 0$, there exist $\gamma(\omega) \geq 0$ and $\pi(\omega)$ such that $\tau(\omega) = \gamma(\omega) \pi(\omega)$, and vice versa.

Thus, we can reformulate (5.35) as follows:

$$\text{CVaR}_\alpha(Q^s(x, \omega), p(x)) = \max \frac{1}{1 - \alpha} \sum_{\omega \in \Omega} \tau(\omega)^\top (r(\omega) - T(\omega)x) \quad (5.36a)$$

$$\text{s.t. } \sum_{\omega \in \Omega} \gamma(\omega) = 1 - \alpha \quad (5.36b)$$

$$\gamma(\omega) \leq p(x, \omega), \quad \forall \omega \in \Omega \quad (5.36c)$$

$$W(\omega)^\top \tau(\omega) - q(\omega)\gamma(\omega) \leq 0, \quad \forall \omega \in \Omega \quad (5.36d)$$

$$\tau(\omega) \in \mathbb{R}_+^{m_2}, \gamma(\omega) \in \mathbb{R}_+, \quad \forall \omega \in \Omega, \quad (5.36e)$$

which is in the form of the min-max problem (5.1).

5.6.2 Two-Stage DRO with Decision-Dependent Ambiguity Set

A two-stage decision-dependent DRO problem is defined as follows:

$$\min_{x \in X} \left\{ f(x) + \max_{p \in \mathcal{P}(x)} \mathbb{E}_p[Q^s(x, \omega)] \right\}, \quad (5.37)$$

where $\mathcal{P}(x)$ is an ambiguity set that depends on x , and $Q^s(x, \omega) = \min\{q(\omega)^\top y : W(\omega)y = r(\omega) - T(\omega)x, y \in \mathcal{Y}\}$ for $\omega \in \Omega$. An example of an ambiguity set is *moment-matching ambiguity set*, where the distribution's moments match the known moment information. Specifically, let $\zeta(\omega) := (\zeta_1(\omega), \zeta_2(\omega), \dots, \zeta_t(\omega))^\top \in \mathbb{R}^t$ be moment functions on Ω . Then, a decision-dependent moment-matching ambiguity set is defined as:

$$\mathcal{P}(x) := \left\{ p \in \mathbb{R}_+^{|\Omega|} : l(x) \leq \sum_{\omega \in \Omega} p(\omega)\zeta(\omega) \leq u(x), \right. \\ \left. \underline{p}(x) \leq p \leq \bar{p}(x), \sum_{\omega \in \Omega} p(\omega) = 1 \right\}. \quad (5.38)$$

Here, $\underline{p}(x) : \mathbb{R}^{n_1} \rightarrow \mathbb{R}^{|\Omega|}$, $\bar{p}(x) : \mathbb{R}^{n_1} \rightarrow \mathbb{R}^{|\Omega|}$, $l(x) : \mathbb{R}^{n_1} \rightarrow \mathbb{R}^t$, and $u(x) : \mathbb{R}^{n_1} \rightarrow \mathbb{R}^t$ are predetermined functions that specify lower and upper bounds for a given x . The DRO problem (5.37) can be seen as a min-max formulation, where the recourse is associated with the decision p . We call the problem (5.37) has *relatively complete* ambiguity set, if $\mathcal{P}(x) \neq \emptyset$ for all $x \in X$. It is important to note that the L^2 methods can address the DRO problem (5.37) even in the absence of the relatively complete ambiguity set. Specifically, in the L^2 methods, by utilizing certain types of cuts, we can cut off infeasible solutions x where $\mathcal{P}(x) = \emptyset$ while running the algorithm. For instance, if the ambiguity set is empty for a solution \hat{x} , then we can cut it off from the feasible region X by adding the following cut:

$$\sum_{i \in I|\hat{x}_i=0} x_i + \sum_{i \in I|\hat{x}_i=1} (1 - x_i) \geq 1. \quad (5.39)$$

This presents a distinct advantage of our approach when compared to an existing approach in the literature that is based on duality results.

In the dual-based approach for (5.37), we dualize the inner maximization problem, using strong duality of some special types of ambiguity sets, to derive a single-level reformulation, so-called a dual reformulation (e.g., see [7, 60, 106]). For the moment ambiguity set (5.38), the dual reformulation of the DRO model (5.37) is given by

$$\min f(x) - \underline{\alpha}^\top l(x) + \bar{\alpha}^\top u(x) - \underline{\beta}^\top \underline{p}(x) + \bar{\beta}^\top \bar{p}(x) \quad (5.40a)$$

$$\text{s.t. } x \in X \quad (5.40b)$$

$$(-\underline{\alpha} + \bar{\alpha})^\top \zeta(\omega) - \underline{\beta}(\omega) + \bar{\beta}(\omega) \geq Q(x, \omega), \quad \forall \omega \in \Omega \quad (5.40c)$$

$$(\underline{\alpha}, \bar{\alpha}, \underline{\beta}, \bar{\beta}) \in \mathbb{R}_+^t \times \mathbb{R}_+^t \times \mathbb{R}_+^{|\Omega|} \times \mathbb{R}_+^{|\Omega|}, \quad (5.40d)$$

where $\underline{\alpha}, \bar{\alpha}, \underline{\beta} = (\underline{\beta}(\omega))_{\omega \in \Omega}^\top$, and $\bar{\beta} = (\bar{\beta}(\omega))_{\omega \in \Omega}^\top$ are dual multipliers for the constraints in (5.38). However, this dual approach presents computational challenges in practice. First, the

dual reformulations rely on relatively complete ambiguity sets, which may be impractical, as demonstrated by our computational results in Section 5.7.3. One might consider using the penalty method—introducing penalties to address violated solutions—but it fails in the DRO problem (5.37) due to its min-max structure. In the inner maximization problem, penalties must be applied negatively; however, these negative penalties can promote violations in the outer minimization problem, rather than prevent them. Another challenge is the scalability of the problem. Specifically, $Q(x, \omega)$ in (5.40c) is typically approximated using valid cuts. As the number of scenarios increase, the decomposed problems become increasingly difficult to solve due to the growing number of cuts. The nonconvex terms in the objective function also present further challenges in solving the decomposed problems. We note that the scalability issue is not limited to this specific type of ambiguity sets. When considering an ambiguity set defined using Wasserstein metric, so-called *Wasserstein ambiguity set*, it is required to add $|\Omega| \times |\Omega|$ cuts in each iteration, readily resulting in a substantially large subproblem; e.g., see the algorithm presented in Duque and Morton [31].

5.6.3 Bi-Parameterized Stochastic Network Interdiction Problem

Recall that the generic formulation of stochastic interdiction problems is given by

$$\min_{x \in X} \mathbb{E} \left[\max_{y \in Y(x, \omega)} f(x, y, \omega) \right].$$

Most studies investigating these problems typically assume that the interdiction x affects either the objective function $f(x, y, \omega)$ or the network user's feasible set $Y(x, \omega)$, but not both, to derive efficient solution approaches (see, e.g., [26, 49, 64, 68]).

The min-max form of BTSP (5.1) generalizes stochastic network interdiction problems by relaxing this assumption, thereby allowing for the modeling of more realistic situations. For

instance, consider a network user seeking the shortest path on a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where feasible paths are subject to resource constraints. These resources can represent any values that may change during travel along an arc, such as travel time, fuel consumption, or load weight. These constraints ensure that the total resource usage along a path either meets or falls within specified thresholds. In this context, the interdicator may disrupt the network user's overall resource system, making it more challenging to satisfy the resource constraints. The resource constraint can be expressed by the following form, where the rhs depends on the interdiction decision x''_k for each resource $k \in K$:

$$\sum_{a \in \mathcal{A}} r_{\omega ka} y_{\omega a} \leq h_{\omega k} + s_{\omega k} x''_k, \quad (5.41)$$

where, for scenario ω , $r_{\omega ka} \in \mathbb{R}$ represents the resource change on arc a , $h_{\omega k} \in \mathbb{R}$ expresses the nominal resource threshold, and $s_{\omega k} \in \mathbb{R}$ denotes the impact of interdiction on the threshold.

Note that the interpretation of these constraints is not limited to resource contexts; for instance, in a surveillance coverage scenario (for the network user), the interdicator could force the network user to pass through specific nodes or arcs, causing a detour to the surveillance destination. Incorporating these constraints into the network user's problem introduces integral restrictions on variables. While it is well-known that the feasible region of the conventional shortest path problem is integral—allowing the problem to be solved using its continuous relaxation without compromising optimality [25]—introducing resource constraints eliminates this integral property. As a result, the network user's problem is required to have the integral restrictions on decision variables, i.e., $y_{\omega a} \in \{0, 1\}$ for all $a \in \mathcal{A}$ and $\omega \in \Omega$. Refer to Section 5.7.2 for results of our computational experiments for bi-parameterized stochastic network interdiction problem.

5.7 Computational Results

We conduct numerical experiments to evaluate the computational efficiency of the proposed approaches. We consider three problem sets: (a) bi-parameterized min-min stochastic and distributionally robust facility location problem, (b) bi-parameterized min-max stochastic network interdiction problem, and (c) distributionally robust facility location problem with decision-dependent ambiguity set. In our implementation of the L^2 methods, σ is set to a certain large value chosen after performing preliminary tests. For the regularized L^2 methods, we simplify the model by dropping the dependency of variables $\mu(\omega)$ on ω , thereby reducing the dimension of solution space and the computational burden. Detailed parameter settings are provided for each specific problem in the ensuing sections. All algorithms were coded in Julia 1.9 and implemented through the branch-and-cut framework of Gurobi 9.5. The optimality tolerance is set to 10^{-4} , and the time limit is set to one hour. We conducted all tests on a machine with an Intel Core i7 processor (3.8 GHz) and 32 GB of RAM, using a single thread.

5.7.1 Bi-Parameterized (Min-Min) Facility Location Problem

We introduce a bi-parameterized facility location problem (BiFLP), where the first-stage decision involves both establishing facilities and contracting outsourcing suppliers. Unlike the traditional facility location problem, customer demand can also be met by outsourcing suppliers, with contracts established in advance to reduce procurement costs. The decision-maker must balance the trade-off between building their own facilities (which incur higher fixed costs but lower variable costs) and outsourcing (which involves lower fixed costs but higher variable costs). In this section, we address both risk-neutral and DRO variants of this

problem as BTSP and DR-BTSP, respectively.

Let $I = \{1, \dots, n_1\}$ be the set of potential facility locations, $J = \{1, \dots, n_2\}$ the set of demand locations, and $K = \{1, \dots, n_3\}$ the set of potential outsourcing suppliers. Binary variables x'_i for $i \in I$ and x''_k for $k \in K$ represent the decision to build a facility at location i and the decision to establish an outsourcing contract with supplier k , respectively, subject to budget $b > 0$. Let $x = (x', x'')$ denote a vector of all first-stage decision variables. The random demand at location $j \in J$ is represented by random variable \tilde{d}_j , and its realizations are denoted by $d_j(\omega)$ for $\omega \in \Omega$. Demand can be fulfilled by both facility at $i \in I$ and supplier $k \in K$. The flow from facility i to demand location j is denoted by variable y_{ij} , and from supplier k to j by variable u_{kj} . The unit transportation costs to j are c_{ij} for facility i and $(q_{kj} - s_{kj}x''_k)$ for supplier k . Here, q_{kj} represents the unit transportation cost from supplier k to demand location j without an outsourcing contract, i.e., $x''_k = 0$. This cost is reduced by $s_{kj} \geq 0$ if a contract is established, i.e., $x''_k = 1$. Let h_i be the capacity of each facility $i \in I$. The first-stage feasible region is defined as $X = \{(x', x'') \in \{0, 1\}^{n_1} \times \{0, 1\}^{n_3} : \kappa_1^\top x' + \kappa_2^\top x'' \leq b\}$, where $\kappa_1 \in \mathbb{R}_+^{n_1}$ and $\kappa_2 \in \mathbb{R}_+^{n_3}$ are cost vectors associated with establishing facilities and outsourcing contracts, respectively.

5.7.1.1 Risk-Neutral BiFLP

The formulation of risk-neutral BiFLP is given by $\min_{x \in X} \sum_{\omega \in \Omega} p(\omega)Q(x, \omega)$ where

$$Q(x, \omega) = \min \sum_{i \in I, j \in J} c_{ij} y_{ij} + \sum_{k \in K, j \in J} (q_{kj} - s_{kj} x_k'') u_{kj} \quad (5.42a)$$

$$\text{s.t. } \sum_{i \in I} y_{ij} + \sum_{k \in K} u_{kj} \geq d_j(\omega), \quad \forall j \in J, \quad (5.42b)$$

$$\sum_{j \in J} y_{ij} \leq h_i x_i', \quad \forall i \in I, \quad (5.42c)$$

$$y \in \mathbb{R}_+^{n_1 \times n_2}, \quad u \in \mathbb{R}_+^{n_3 \times n_2}, \quad (5.42d)$$

for $\omega \in \Omega$. The objective (5.42a) is to minimize the total cost of fulfilling demand, considering both transportation costs from facilities and outsourcing suppliers. Constraints (5.42b) ensure that demand at all locations $j \in J$ are satisfied. Constraint (5.42c) for each $i \in I$ limits the total flow from facility i by its capacity h_i .

To generate test instances, we randomly place $(n_1 + n_2 + n_3)$ points on a 100×100 grid, representing potential facility locations, demand locations, and supplier locations. We consider four network sizes, with (n_1, n_2, n_3) set to $(12, 40, 5)$, $(12, 40, 10)$, $(20, 60, 5)$, and $(20, 60, 10)$. The costs of building a facility $\kappa_{1i} = 5$ and contracting an outsourcing supplier $\kappa_{2k} = 4$ for all $i \in I$ and $k \in K$, and the budget $b = (5n_1 + 4n_3)/4$. The cost of fulfilling demand at location $j \in J$ from supplier $k \in K$ consists of a fixed component and a distance-dependent component: specifically, $q_{kj} = \bar{c} + 2v(k, j)$ and $s_{kj} = \bar{c} + 0.8v(k, j)$, where \bar{c} is a predetermined fixed cost, and $v(k, j)$ is the Euclidean distance between k and j . Each facility at $i \in I$ has a capacity of $h_i = 100$. Demand data are generated using normal distributions. For instances with $n_3 = 5$, the mean demand $\bar{\mu}_j$ for each $j \in J$ is uniformly drawn from $\{40, 41, \dots, 80\}$, and for instances with $n_3 = 10$, it is drawn from $\{50, 51, \dots, 90\}$. In both cases, the standard

deviation for each $j \in J$ is set to $\bar{\mu}_j/4$.

For benchmark comparisons, we consider two approaches: an approach akin to the integer L-shaped method (IL) and a deterministic expanded formulation (DE). In IL, the recourse function is approximated using integer optimality cuts, as described in Proposition 2 in Laporte and Louveaux [55]. Unlike the standard integer L-shaped method, IL does not generate continuous optimality cuts since the continuous relaxation of the recourse problem (5.2) does not provide dual information for generating such cuts. For the min-min problem, DE is formulated as a large-scale mixed-integer bilinear program:

$$\min \left\{ \sum_{\omega \in \Omega} p(\omega) \left(\sum_{i \in I, j \in J} c_{ij} y_{ij} + \sum_{k \in K, j \in J} (q_{kj} - s_{kj} x_k'') u_{kj} \right) : x \in X, (5.42b)-(5.42d), \forall \omega \in \Omega \right\}.$$

In our tests, this formulation is solved directly using Gurobi 9.5, with *NonConvex* parameter set to 2. In the standard L^2 method (denoted by L^2), we set $\sigma_i = 10^5$ for all $i \in I$. In the regularized L^2 method (denoted by L^2 -R), we scale the objective by a factor of 10^{-2} to balance its magnitude with the regularization term. Additionally, we set $\gamma = 10^{-4}$ and $\bar{u}_i = 10^3$ for all $i \in I$ and define the regularization function as $R(\mu) = \sum_{i \in I} \mu_i$.

Table 5.1 summarizes the test results for the BiFLP instances. Each row presents the average results of three instances with the same network structure, (n_1, n_2, n_3) , and the same number of scenarios, $|\Omega|$. The columns labeled “Gap (%)” and “Time (s)” report the optimality gap (in %) and solution time (in seconds), respectively. The optimality gap results are marked as “NA” for instances where an algorithm failed to find both primal and dual bounds within the time limit. The results show that L^2 outperforms the other approaches in terms of the computational efficiency. On average, L^2 is 17.6 times faster than IL and is 7.4 times faster than DE for the instances solved to optimality by all approaches. These factors increase to 21.9 and 8.8 times, respectively, when considering all instances. The IL showed

Table 5.1: Results for BiFLP instances.

Instance		L^2		L^2 -R		DE		IL	
(n_1, n_2, n_3)	$ \Omega $	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)
(12, 40, 5)	10	0.0	2	0.0	2	0.0	2	0.0	20
	50	0.0	6	0.0	7	0.0	27	0.0	78
	100	0.0	12	0.0	12	0.0	95	0.0	142
	200	0.0	23	0.0	25	0.0	344	0.0	296
	500	0.0	49	0.0	54	0.0	1732	0.0	652
	1000	0.0	131	0.0	131	NA	3600+	0.0	1300
(12, 40, 10)	10	0.0	18	0.0	33	0.0	20	0.0	469
	50	0.0	60	0.0	67	0.0	320	0.0	1320
	100	0.0	137	0.0	148	0.0	1168	0.0	2373
	200	0.0	196	0.0	243	NA	3600+	100.0	3600+
(20, 60, 5)	10	0.0	40	0.0	69	0.0	8	100.0	3600+
	50	0.0	110	0.0	126	0.0	171	100.0	3600+
	100	0.0	113	0.0	112	0.0	457	100.0	3600+
	200	0.0	263	0.0	290	0.0	1399	100.0	3600+
(20, 60, 10)	10	0.0	285	0.0	394	0.0	92	100.0	3600+
	50	0.0	854	0.0	867	0.0	1697	100.0	3600+
	100	0.0	1240	0.0	1422	NA	3600+	100.0	3600+
	200	5.0*	2423*	6.8**	2612**	NA	3600+	100.0	3600+

* Average over three instances: (1) 0% gap, 1909 s, (2) 0% gap, 1756 s, and (3) 14.9% gap, 3600+ s.

** Average over three instances: (1) 0% gap, 2057 s, (2) 0% gap, 2175 s, and (3) 20.5% gap, 3600+ s.

poor scalability due to its limited capability in improving dual bounds; specifically, for the instances with $(n_1, n_2, n_3) = (20, 60, 5)$ and $(20, 60, 10)$, IL could not reduce optimality gaps within the time limit for all instances. We find that DE's performance is less sensitive to the network size than the others, but it is significantly affected by the number of scenarios. For the first instance category, with $(12, 40, 5)$ network and 10 scenarios, DE and L^2 solved instances in similar solution times. However, as the number of scenarios increases to 500, DE's solution time increases by around 900 times, while L^2 's solution time increases only by around 26 times. When comparing the results from L^2 and L^2 -R, the performance differences are minor in terms of solution time. The standard L^2 method is, on average, 1.1 times faster than the regularized L^2 method for instances where both methods solved to optimality.

5.7.1.2 Distributionally Robust BiFLP

We now consider the DRO variant of BiFLP, denoted by DR-BiFLP, which is formulated as the following DR-BTSP: $\min_{x \in X} \max_{p \in \mathcal{P}} \sum_{\omega \in \Omega} p(\omega) Q(x, \omega)$ where, for $\omega \in \Omega$, $Q(x, \omega)$ is given by (5.42). Here, the ambiguity set \mathcal{P} is defined as the moment-matching ambiguity set (5.38), which is independent of the decision x , i.e., the parameters $\underline{p}(x)$, $\bar{p}(x)$, $l(x)$, and $u(x)$ are vectors that do not vary with x . We generate the test instances using the same configurations of (n_1, n_2, n_3) and Ω , as the risk-neutral instances, but with different random seeds. For the comparison, we consider a dual-based approach, denoted by DA-DE. In this approach, we take the dual of the inner maximization (as a special case of (5.40)), reformulate the problem into a single-level deterministic extended form, by integrating the recourse problem into the constraints (5.40c), and solve it directly. In our preliminary tests, we observed minor differences in the results from the standard and regularized L^2 methods. Therefore, we report here only the results from the standard L^2 method. Each row of Table 5.2 presents the average result of three instances within the corresponding instance category. The results indicate that, on average, the L^2 method achieves optimal solutions 1.7 times faster compared to the dual-based approach.

5.7.2 Bi-Parameterized (Min-Max) Network Interdiction Problem

Next, we consider a bi-parameterized (min-max) network interdiction problem (BiNIP). Consider a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N} = \{1, \dots, n_1\}$ is the set of nodes and $\mathcal{A} = \{1, \dots, n_2\}$ is the set of arcs in the graph. Resources that restrict each path in this network is indexed by $K = \{1, \dots, n_3\}$. Variable $x'_a \in \{0, 1\}$ for each $a \in \mathcal{A}$ indicates whether arc $a \in \mathcal{A}$ is interdicted. For resources, $x''_k \in \{0, 1\}$ for $k \in K$ represents whether interdiction occurs for resource k or not. Let $x = (x', x'')$ represents a vector of all interdiction

Table 5.2: Results for DR-BiFLP instances.

Instance		L^2		DA-DE	
(n_1, n_2, n_3)	$ \Omega $	Gap (%)	Time (s)	Gap (%)	Time (s)
(12, 40, 5)	10	0.0	2	0.0	4
	50	0.0	5	0.0	25
	100	0.0	10	0.0	69
	200	0.0	19	0.0	222
	500	0.0	36	0.0	304
	1000	0.0	83	0.0	1508
(12, 40, 10)	10	0.0	15	0.0	4
	50	0.0	76	0.0	85
	100	0.0	95	0.0	233
	200	0.0	189	0.0	650
(20, 60, 5)	10	0.0	33	0.0	12
	50	0.0	59	0.0	198
	100	0.0	121	0.0	306
	200	0.0	199	0.0	1177
(20, 60, 10)	10	0.0	351	0.0	28
	50	0.0	694	0.0	180
	100	0.0	1413	0.0	1357
	200	0.0	1620	1.5*	2484*

* Average over three instances: (1) 0% gap, 2597 s, (2) 0% gap, 1254 s, and (3) 4.4% gap, 3600+ s.

decision variables. The interdiction decisions are associated with costs $\kappa_a \in \mathbb{R}_+$ for arcs and $g_k \in \mathbb{R}_+$ for resources, and the total cost is constrained by budget $b \in \mathbb{R}_+$. The first-stage feasible region is defined as $X = \{(x', x'') \in \{0, 1\}^{n_2} \times \{0, 1\}^{n_3} : \kappa^\top x' + g^\top x'' \leq b\}$. Variable $y_a \in \{0, 1\}$ for each $a \in \mathcal{A}$ represents whether the network user traverses arc a ($y_a = 1$) or not ($y_a = 0$). Using random variable \tilde{d}_a , we represent the increase in arc length due to interdiction for each $a \in \mathcal{A}$. Its realizations are denoted by $d_a(\omega)$ for $\omega \in \Omega$. The length of arc a for scenario ω becomes $(c_a + d_a(\omega)x'_a)$, where c_a is the nominal length of arc a when not interdicted. The change in resource $k \in K$ after traversing arc $a \in \mathcal{A}$ is denoted by r_{ka} , and the threshold is denoted by h_k . When interdiction occurs, i.e., $x''_k = 1$, this threshold is adjusted by s_k . The formulation of BiNIP is given by $\max_{x \in X} \sum_{\omega \in \Omega} p(\omega) Q(x, \omega)$ where, for

$\omega \in \Omega$,

$$Q(x, \omega) = \min \sum_{a \in \mathcal{A}} (c_a + d_a(\omega)x'_a)y_a \quad (5.43a)$$

$$\text{s.t. } Ty = q, \quad (5.43b)$$

$$\sum_{a \in \mathcal{A}} r_{ka}y_a \geq h_k + s_kx''_k, \quad \forall k \in K, \quad (5.43c)$$

$$y \in \{0, 1\}^{n_2}. \quad (5.43d)$$

The first-stage problem aims to maximize the expected path length, with interdiction solutions restricted by the cardinality constraint in X . In the network user's problem, the objective function (5.43a) represents the length of the path. Constraints (5.43b) enforce the balance of incoming and outgoing flows for each node; $T \in \{-1, 0, 1\}^{n_1 \times n_2}$ is the node-arc incidence matrix, where $T_{ia} = 1$ if arc $a \in \mathcal{A}$ leaves node $i \in \mathcal{N}$, $T_{ia} = -1$ if arc a enters node i , and $T_{ia} = 0$ otherwise. Also, $q \in \{-1, 0, 1\}^{n_2}$ is a vector where $q_i = 1$ if $i \in \mathcal{N}$ is the source node, $q_i = -1$ if i is the sink node, and $q_i = 0$ otherwise. Constraints (5.43c) are the resource constraints. Notably, the integral restrictions (5.43d) on y are necessary—unlike in the conventional shortest path problem—since the resource constraints may eliminate the integral property of the feasible region.

In our experiments, we use randomly generated instances based on instances from Nguyen and Smith [68]. We utilize their data on network topology, arc lengths, and deterministic penalty lengths. We consider two categories of their instances: 20-node and 40-node instances, with 10 instances in each category. The number of arcs varies in [59, 78] for the 20-node instances, and [277, 306] for the 40-node instances. We extend these instances by introducing random penalty lengths, which are sampled from a uniform distribution over the interval $[\bar{d}_a - o_a, \bar{d}_a + o_a]$ for each arc $a \in \mathcal{A}$, where \bar{d}_a is the deterministic penalty length from Nguyen and Smith [68]. The offset $o_a = 3$ for the 20-node instances and $o_a = 4$ for the

Table 5.3: Results for BiNIP instances.

Instance			L^2			IL		
n_1	n_2	n_3	$ \Omega $	Gap (%)	Time (s)	Gap (%)	RelGap (%)	Time (s)
20	[59, 78]	3	10	0.0	0.6	100.0	3.6	3600+
			20	0.0	1.4	100.0	4.0	3600+
			50	0.0	3.6	100.0	6.8	3600+
			100	0.0	7.2	100.0	6.4	3600+
			500	0.0	41.7	100.0	11.2	3600+
			1000	0.0	60.9	100.0	9.5	3600+
40	[277, 306]	4	10	0.0	5.8	100.0	3.6	3600+
			20	0.0	10.8	100.0	4.0	3600+
			50	0.0	31.6	100.0	4.3	3600+
			100	0.0	63.6	100.0	5.3	3600+

40-node instances. For the 20-node instances, we set budget $b = 4$, arc interdiction costs $\kappa_a = 1$ for all $a \in \mathcal{A}$, and resource interdiction costs $g_k = 2$ for all $k \in K$. We consider three resources ($K = \{1, 2, 3\}$). The resource consumption parameter r_{ka} is randomly drawn from $\{1, 2\}$ for each $k \in K$ and $a \in \mathcal{A}$. The threshold vector $h = (5, 4, 3)^\top$, and the penalty vector $s = (1, 2, 3)^\top$. Similarly, for the 40-node instances, budget b is set to 5 with the same arc and resource interdiction costs. We consider four resources ($K = \{1, 2, 3, 4\}$), where r_{ka} is randomly drawn from $\{1, 2, 3\}$ for each $k \in K$ and $a \in \mathcal{A}$. We set the threshold vector $h = (10, 8, 6, 4)^\top$ and the penalty vector $s = (1, 3, 5, 7)^\top$.

To benchmark the proposed approach, we consider IL for BiNIP. Note that DE is not applicable to BiNIP due to its min-max form. In preliminary tests, we found minor differences between the outcomes of L^2 and L^2 -R. Therefore, we report only the results obtained by L^2 for BiNIP in Table 5.3. For all tests, the parameters σ_k are set to 10^3 for $k \in K$. The test results are summarized in Table 5.3 where each row presents the average results for 10 instances. For each column labeled “ L^2 ” or “IL”, we report the optimality gap (in %) under “Gap (%)”, and the solution time (in seconds) under “Time (s)”. The results under the “RelGap (%)” column represent the relative gaps between the primal bounds obtained

by IL to the optimal objective values. The results show that the L^2 method outperforms IL across all instances. The IL was unable to reduce the dual bounds for all 100 instances within the time limit of 3600 seconds, resulting in 100% optimality gaps, while L^2 found optimal solutions for all instances within 23 seconds on average. When comparing primal bounds, IL produced primal bounds that were, on average, 5.7% worse than those obtained by the L^2 method, even if the former spent 158 times more computational time.

5.7.3 Distributionally Robust Facility Location with Decision-Dependent Ambiguity Set

Lastly, to showcase the applications of BTSPs for tackling decision-dependent uncertainty, we consider a distributionally robust two-stage facility location problem under decision-dependent demand uncertainty (DRFLP), which is a modified version of the problem presented in Yu and Shen [106]. In this problem, locations of facilities impact service accessibility, thereby affecting the realizations of random demand. While the exact distribution of demand is unknown, we model its moments as functions of chosen locations to express how demand depends on these decisions. Specifically, these functions in our model capture the relationship where locating a facility closer to a demand point increases its mean demand more than placing it farther away. To determine location decisions that are robust under this demand uncertainty and distributional ambiguity, we employ a DRO model, where the ambiguity set is defined using these functions that represent the moment information.

Let $I := \{1, \dots, n_1\}$ denote the set of potential facility locations, and $J := \{1, \dots, n_2\}$ denote the set of demand locations. The decision to establish a facility at location $i \in I$ is represented by binary variable x_i , where $x_i = 1$ indicates building a facility at location i , and $x_i = 0$ otherwise. The total number of facilities is constrained by a budget $b > 0$.

The random demand at each demand location $j \in J$ is denoted by \tilde{d}_j , and its realization is denoted by $d_j(\omega)$ for $\omega \in \Omega$. The flow decision from facility $i \in I$ to demand location $j \in J$ is represented by y_{ij} . The unit transportation cost for flow between facility i and demand location j is c_{ij} . If demand at location $j \in J$ is not fully satisfied, a penalty cost q_j is caused for each unit of the unsatisfied demand. Additionally, the capacity of each facility $i \in I$ is denoted by h_i , which limits the total flow emanating from facility i . The formulation of DRFLP is given by

$$\min_{x \in X} \max_{p \in \mathcal{P}(x)} \sum_{\omega \in \Omega} p(\omega) Q(x, \omega)$$

where $X = \{x \in \{0, 1\}^{n_1} : \sum_{i \in I} x_i \leq b\}$ and for $\omega \in \Omega$,

$$Q(x, \omega) = \min \sum_{i \in I, j \in J} c_{ij} y_{ij} + \sum_{j \in J} q_j u_j \quad (5.44a)$$

$$\text{s.t. } \sum_{i \in I} y_{ij} + u_j \geq d_j(\omega), \quad \forall j \in J \quad (5.44b)$$

$$\sum_{j \in J} y_{ij} \leq h_i x_i, \quad \forall i \in I \quad (5.44c)$$

$$y \in \mathbb{R}_+^{n_1 \times n_2}, \quad u \in \mathbb{R}_+^{n_2}. \quad (5.44d)$$

The objective of the recourse problem is to minimize the sum of transportation and penalty costs as described in (5.44a). Constraints (5.44b) ensure that the total flow into demand location $j \in J$, along with the amount u_j , satisfies the demand $d_j(\omega)$. Constraints (5.44c) limits the total flow from each facility $i \in I$ by its capacity h_i if the facility is established (i.e., $x_i = 1$), or to zero otherwise. The dual formulation of the recourse problem is given as

follows:

$$Q(x, \omega) = \max \sum_{j \in J} d_j(\omega) \pi_j - \sum_{i \in I} h_i x_i \nu_i \quad (5.45a)$$

$$\text{s.t. } \pi_j - \nu_i \leq c_{ij}, \quad \forall i \in I, j \in J \quad (5.45b)$$

$$\pi_j \leq q_j, \quad \forall j \in J \quad (5.45c)$$

$$\pi \in \mathbb{R}_+^{n_2}, \nu \in \mathbb{R}_+^{n_1}. \quad (5.45d)$$

We utilize this dual formulation in all tests. The ambiguity set $\mathcal{P}(x)$ is defined as a moment-matching ambiguity set, i.e.,

$$\begin{aligned} \mathcal{P}(x) = \left\{ p \in \mathbb{R}_+^{|\Omega|} : \sum_{\omega \in \Omega} p(\omega) = 1, \right. \\ \sum_{\omega \in \Omega} p(\omega) d_j(\omega) \geq M_j(x) - \epsilon_j^M, \quad \forall j \in J, \\ \sum_{\omega \in \Omega} p(\omega) d_j(\omega) \leq M_j(x) + \epsilon_j^M, \quad \forall j \in J, \\ \sum_{\omega \in \Omega} p(\omega) (d_j(\omega))^2 \geq S_j(x) \underline{\epsilon}_j^S, \quad \forall j \in J \\ \left. \sum_{\omega \in \Omega} p(\omega) (d_j(\omega))^2 \leq S_j(x) \bar{\epsilon}_j^S, \quad \forall j \in J \right\}. \end{aligned} \quad (5.46)$$

This ambiguity set consists of bounding constraints on the first and second moments of \tilde{d}_j for $j \in J$ under the probability distribution p . The parameters are defined as follows:

$$\begin{aligned} M_j(x) &= \bar{\mu}_j \left(1 + \sum_{i \in I} \rho_{ij}^M x_i \right), \quad \forall j \in J \\ S_j(x) &= \left(\bar{\mu}_j^2 + \bar{\sigma}_j^2 \right) \left(1 + \sum_{i \in I} \rho_{ij}^S x_i \right), \quad \forall j \in J. \end{aligned}$$

Here, $\bar{\mu}_j$ and $\bar{\sigma}_j$ are the baseline first and second moments for $j \in J$. Parameters $\rho_{ij}^M = e^{-v(i,j)/25}$ and $\rho_{ij}^S = e^{-v(i,j)/50}$ represent the impact of building facility at i on the moments

of \tilde{d}_j , where $v(i, j)$ is the Euclidean distance between locations $i \in I$ and $j \in J$. We set $\epsilon_j^M = 25$, $\epsilon_j^S = 0.1$, and $\bar{\epsilon}_j^S = 1.9$ for all $j \in J$.

To generate test instances, we place $(n_1 + n_2)$ random locations on a 100×100 grid for facility and demand locations. The transportation cost c_{ij} is set to the Euclidean distance $v(i, j)$ from location $i \in I$ to location $j \in J$. We set the capacity $h_i = h = 500$ for $i \in I$. The mean values $\bar{\mu}_j$ for $j \in J$ are uniformly sampled from $\{\underline{d}_j, \underline{d}_j + 1, \dots, \bar{d}_j\}$, where \underline{d}_j is the nearest integer to $(0.7 \times b \times h/n_2)$ and \bar{d}_j is the nearest integer to $\text{round}(b \times h/n_2)$. Then, we sample the demand realizations $d_j(\omega)$ for $\omega \in \Omega$ from $N(\bar{\mu}_j, 0.8\bar{\mu}_j)$, truncated within $[1, 300]$, where $0.8\bar{\mu}_j$ is the standard deviation.

In the L^2 methods for DRFLP, we include an additional step that determines the worst-case distribution within the ambiguity set after solving subproblems. In particular, this step involves solving the *distribution separation problem*, defined as follows:

$$\max_{p \in \mathcal{P}(z), z \in X} \sum_{\omega \in \Omega} p(\omega) \hat{Q}^l(x^l, \omega) + \sum_{i \in I} \lambda_i^l z_i \quad (5.47)$$

where $\lambda_i^l = \sigma_i(2x_i^l - 1)$ in Algorithm 5.1 or $\lambda_i^l = \mu_i^l(2x_i^l - 1)$ in Algorithm 5.3, respectively. We denote the worst-case distribution identified by solving the distribution separation problem in iteration l by $p^l = (p^l(\omega))_{\omega \in \Omega}^\top$. Using p^l , we evaluate the objective $\phi^l = f(x^l) - \sigma^\top x^l + \sum_{\omega \in \Omega} p^l(\omega) \hat{Q}^l(x^l, \omega)$ and obtain an optimality cut in the following form:

$$\theta \geq \sum_{\omega \in \Omega} p^l(\omega) \left(\sum_{j \in J} d_j(\omega) \pi_j^l(\omega) - \sum_{i \in I} h_i x_i \nu_i^l(\omega) \right) + \sum_{i \in I} \sigma_i (2x_i - 1) z_i^l,$$

where $(\pi^l(\omega), \nu^l(\omega))$ is an optimal solution for the subproblem for scenario ω at iteration l . To reduce the computational burden, we fix z to x^l in the distribution separation problem. Note that the cut generated by its solution with $z = x^l$ is valid for $\max_{p \in \mathcal{P}(x)} \mathbb{E}[Q(x, \omega)]$, as discussed in Remark 5.2. If the ambiguity set is empty for a given current solution, i.e.,

Table 5.4: Results from L^2 -R and DA for the DRFLP instances.

Instance		L^2 -R			DA		
(n_1, n_2, b)	$ \Omega $	Gap (%)	Time (s)	#OptCut	Gap (%)	Time (s)	#OptCut
(15, 20, 4)	500	0.0	34	101	0.0	39	47013
	1000	0.0	37	86	Unbounded		
	2000	0.0	48	61	0.0	76	94087
	5000	0.0	225	118	0.0	621	583829
	10000	0.0	578	140	0.0	3589	1552667
(20, 20, 4)	500	0.0	27	104	0.0	38	48476
	1000	0.0	65	143	0.0	212	131379
	2000	0.0	144	177	0.0	565	322976
	5000	0.0	515	222	0.0	2178	849847
	10000	0.0	844	156	22.6	3600+	1298870
(30, 20, 4)	500	0.0	379	937	Unbounded		
	1000	0.0	536	885	21.4	3600+	644380
	2000	0.0	320	276	0.0	3273	514143
	5000	0.0	1228	457	Unbounded		
	10000	0.0	1683	286	100.0	3600+	932400
(40, 20, 4)	500	0.0	492	989	Unbounded		
	1000	0.0	860	1132	Unbounded		
	2000	0.0	677	459	Unbounded		
	5000	7.9	3600+	1125	72.8	3600+	805440
	10000	0.0	1926	259	Unbounded		

$\mathcal{P}(x^l) = \emptyset$, then we add the following feasibility cut:

$$\sum_{i \in I | x_i^l = 0} x_i + \sum_{i \in I | x_i^l = 1} (1 - x_i) \geq 1.$$

Table 5.4 presents the test results comparing the performance of the L^2 method and the dual-based approach (DA). Here, we focus on the regularized L^2 method (L^2 -R, Algorithm 5.3), as it showed a better performance in our preliminary tests for DRFLP. For L^2 -R, we set $\gamma = 10^{-1}$ and $\bar{u}_i = 10^4$ for all $i \in I$ and define the regularization function as $R(\mu) = \|\mu\|_2^2$. The DA solves the dual reformulation (5.40) of DRFLP using the decomposition algorithm presented in [106]. We consider different instance settings by varying the number of facility locations n_1 in $\{15, 20, 30, 40\}$, and the number of scenarios $|\Omega|$ in $\{500, 1000, 2000, 5000, 10000\}$. We

Table 5.5: Results from L^2 and L^2 -R for DRFLP instances.

Instance		L^2		L^2 -R	
(n_1, n_2, b)	$ \Omega $	Gap (%)	Time (s)	Gap (%)	Time (s)
(20, 20, 4)	100	0.0	162	0.0	14
	300	0.0	324	0.0	36
	500	0.0	620	0.0	27
	1000	0.0	1125	0.0	65
	2000	0.0	2405	0.0	144
	5000	51.7	3600+	0.0	515
	10000	63.1	3600+	0.0	844

fix the number of demand locations $n_2 = 20$ and the budget $b = 4$.

Each row in Table 5.4 reports the optimality gap “Gap (%)”, solution time in seconds “Time (s)”, and the number of optimality cuts “#OptCut” for each instance. Out of the total 20 instances, L^2 -R solves 19 instances to optimality within the time limit, while DA solves only nine instances. DA reports “unbounded” for instances where the ambiguity sets are not relative complete, whereas L^2 -R successfully handles these instances. When comparing solution time, L^2 -R is on average 5.3 times faster than DA for instances that are solved to optimality by both methods. As the number of scenarios increases, the difference in the number of optimality cuts increases, as DA requires significantly more cuts. On average, the L^2 method achieves optimality by adding only 0.04% of the number of cuts generated/used by DA.

Next, we compare the performance of the standard L^2 method (L^2 , Algorithm 5.1) with the regularized L^2 method (L^2 -R, Algorithm 5.3) in Table 5.5. Parameters σ_i for $i \in I$ are set to 10^3 . Test instances have $(n_1, n_2, b) = (20, 20, 4)$ and the number of scenarios $|\Omega| \in \{100, 300, 500, 1000, 2000, 5000, 10000\}$. The results show that L^2 -R is computationally efficient than L^2 for solving the DRFLP instances. The L^2 -R solves all instances, while L^2 is unable to solve the instances with 5000 and 10000 scenarios. For instances where both methods solve to optimality, L^2 -R is, on average, 16.2 times faster than L^2 method.

5.8 Conclusion

In this chapter, we introduced Lagrangian-integrated L-shaped (L^2) methods for solving bi-parameterized two-stage stochastic (min-max and min-min) integer programs (BTSPs), which are applicable to interdiction models and a wide range of optimization problems with decision-dependent uncertainty. For cases where the first-stage decisions are pure binary, we developed two exact algorithms applied to the min-min and min-max BTSPs. Additionally, we proposed a regularization-augmented method to address BTSPs with mixed-integer first-stage decisions. We further extended the L^2 method to tackle distributionally robust optimization (DRO) variants of BTSPs (DR-BTSPs) with finite or continuous support. To evaluate the L^2 method's efficiency, extensive numerical tests were conducted under various settings: (1) a risk-neutral setting for bi-parameterized network interdiction and facility location problems; (2) a distributionally robust setting for bi-parameterized facility location with decision-independent ambiguity set; (3) distributionally robust facility location problem with decision-dependent ambiguity set that might not be relatively complete. The results demonstrated the superior efficiency of the L^2 method compared to benchmark approaches. Specifically, the results showed that our approach converged to optimal solutions of all tested instances of bi-parameterized network interdiction problem within 23 seconds on average, whereas the benchmark method failed to converge for any instance within 3600 seconds. The L^2 method achieved optimal solutions, on average, 18.4 times faster for the risk-neutral setting and 1.7 times faster for the decision-independent DRO setting. For the decision-dependent DRO setting, the L^2 method effectively solved instances with the non-relatively complete ambiguity sets and achieved solutions 5.3 faster than the existing dual-based approach.

Chapter 6.

Conclusion

In this dissertation, we introduced the theoretical and algorithmic foundations of distributionally risk-receptive (DRR) and distributionally robust (DRO) optimization, with a particular focus on stochastic integer programs and network interdiction problems (NIPs) under uncertainty and distributional ambiguity. Motivated by real-world challenges where distributional information is limited, we explored the DRR framework alongside the well-studied DRO paradigm. Our work is particularly relevant for adversarial settings, such as attacker-defender games, where analysis on conflicting perspectives may require considering different risk attitudes of the decision-makers.

Chapter 3 addressed single-round NIPs formulated as two-stage stochastic programs. We introduced DRO and DRR variants for the classical shortest path interdiction problem, which model distributionally risk-averse and risk-receptive attitudes, respectively. In this chapter, we developed exact and approximation algorithms, based on cutting plane and reformulation-based techniques. We proved finite convergence of these algorithms and demonstrated, through extensive computational experiments, that the cutting plane-based algorithms are significantly faster than the reformulation-based algorithms. Our computational study highlighted how DRO models yield robust interdiction strategies, and DRR models identify critical network components from the perspective of the network user.

Chapter 4 extended these ideas to multistage stochastic integer and disjunctive programs and multi-round NIPs. We developed convex approximations of the nonconvex value functions—

using both cutting plane and reformulation techniques—for DRR and DRO models in the multistage setting, also accommodating both decision-independent and decision-dependent uncertainty. The finite convergence results were established for the proposed algorithms for a general class of ambiguity sets. Numerical results on multistage maximum flow and facility location interdiction problems demonstrated that the cutting plane approximations significantly outperform the reformulation-based approximations. Additionally, out-of-sample tests showcased the importance of the DRR models for vulnerability analysis and decision-making under data corruption.

Chapter 5 focused on bi-parameterized two-stage stochastic programs under both min-min and min-max structures, with applications to general NIPs and several classes of stochastic integer programs. We proposed exact and regularization-augmented Lagrangian-integrated L-shaped (L^2) methods. In particular, we demonstrated that the L^2 methods effectively handle DRO problems with decision-dependent ambiguity sets that may be empty for some first-stage decisions. We showed that the proposed methods outperformed existing methods in computational efficiency.

We suggest several directions for future research. First, an important extension involves applying the algorithms to cases where value functions are defined on general mixed-integer sets. In these settings, integrating Lagrangian-based decomposition or recently developed cutting plane techniques—e.g., [22, 98], which have been applied to general stochastic mixed-integer programs—could be promising directions. Another direction is to explore applications of these frameworks in multi-agent systems. While this dissertation focused primarily on two-player games, richer game-theoretic models—such as multi-follower or general leader-follower games—remain underexplored within the DRR and DRO frameworks. Lastly, a case study on complex real-world systems would be a promising direction, potentially uncovering new challenges in methodologies; for example, contingency analysis for power systems often

involves mixed-integer nonlinear models, which beyond the scope of this dissertation.

In conclusion, this dissertation contributes to building a foundation for modeling, solving, and analyzing DRR and DRO stochastic integer programs and interdiction problems, thereby providing a fertile ground for further research on resilient, robust, and effective decision-making in uncertain and adversarial settings.

Bibliography

- [1] Shabbir Ahmed, Filipe Goulart Cabral, and Bernardo Freitas Paulo da Costa. Stochastic Lipschitz dynamic programming. *Mathematical Programming*, 191(2):755–793, 2022. doi: 10.1007/s10107-020-01569-z.
- [2] Egon Balas. *Disjunctive programming*. Springer, 2018. ISBN 3-030-00148-2.
- [3] Egon Balas, Sebastián Ceria, and Gérard Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0–1 programs. *Mathematical Programming*, 58(1):295–324, 1993. doi: 10.1007/BF01581273.
- [4] 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. doi: 10.1016/j.ejor.2019.05.033.
- [5] 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. doi: 10.1007/s10898-020-00986-w.
- [6] 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. doi: 10.1137/17m1115046.
- [7] Beste Basciftci, Shabbir Ahmed, and Siqian Shen. Distributionally robust facility location problem under decision-dependent stochastic demand. *European Journal of Operational Research*, 292(2):548–561, 2021. doi: 10.1016/j.ejor.2020.11.002.
- [8] N. Orkun Baycık and Kelly M. Sullivan. Robust location of hidden interdictions on

- a shortest path network. *IIEE Transactions*, 51(12):1332–1347, 2019. doi: 10.1080/24725854.2019.1597316.
- [9] Halil Bayrak and Matthew D. Bailey. Shortest path network interdiction with asymmetric information. *Networks*, 52(3):133–140, 2008. doi: 10.1002/net.20236.
- [10] Güzin Bayraksan and David K. Love. Data-driven stochastic programming using phi-divergences. In *The Operations Research Revolution*, pages 1–19. INFORMS TutORials in Operations Research, 2015.
- [11] Güzin Bayraksan, Francesca Maggioni, Daniel Faccini, and Ming Yang. Bounds for multistage mixed-integer distributionally robust optimization. *SIAM Journal on Optimization*, 34(1):682–717, 2024. doi: 10.1137/22M147178X.
- [12] Mokhtar S Bazaraa, John J Jarvis, and Hanif D Sherali. *Linear programming and network flows*. John Wiley & Sons, Hoboken, NJ, 2008. ISBN 8126518928.
- [13] Michael GH Bell, U. Kanturska, J.-D. Schmöcker, and Achille Fonzone. Attacker–defender models and road network vulnerability. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1872):1893–1906, 2008. doi: 10.1098/rsta.2008.0019.
- [14] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962. doi: 10.1007/BF01386316.
- [15] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena scientific, Belmont, Mass, 3rd ed edition, 2016. ISBN 978-1-886529-05-2.
- [16] John R. Birge and Francois V. Louveaux. A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research*, 34(3):384–392, 1988.

- [17] Jose Blanchet, Jiajin Li, Sirui Lin, and Xuhui Zhang. Distributionally robust optimization and robust statistics. *arXiv preprint arXiv:2401.14655*, 2024.
- [18] Jose Blanchet, Jiajin Li, Markus Pelger, and Greg Zanotti. Automatic outlier rectification via optimal transport. *arXiv preprint arXiv:2403.14067*, 2024.
- [19] Immanuel M. Bomze, Markus Gabl, Francesca Maggioni, and Georg Ch Pflug. Two-stage stochastic standard quadratic optimization. *European Journal of Operational Research*, 299(1):21–34, 2022. doi: 10.1016/j.ejor.2021.10.056.
- [20] Gerald Brown, Matthew Carlyle, Javier Salmerón, and Kevin Wood. Defending critical infrastructure. *Interfaces*, 36(6):530–544, 2006. doi: 10.1287/inte.1060.0252.
- [21] Junyu Cao and Rui Gao. Contextual decision-making under parametric uncertainty and data-driven optimistic optimization. *Optimization Online preprint:2021/10/8634*, 2021.
- [22] Claus C. Carøe and Rüdiger Schultz. Dual decomposition in stochastic integer programming. *Operations Research Letters*, 24(1-2):37–45, 1999. doi: 10.1016/S0167-6377(98)00050-9.
- [23] Claus C. Carøe and Jørgen Tind. A cutting-plane approach to mixed 0–1 stochastic integer programs. *European Journal of Operational Research*, 101(2):306–316, 1997. doi: 10.1016/S0377-2217(96)00399-2.
- [24] 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. doi: 10.1111/j.1467-8306.2004.00410.x.

- [25] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. *Integer programming*, volume 271. Springer, 2014. ISBN 331911008X.
- [26] Kelly J. Cormican, David P. Morton, and R. Kevin Wood. Stochastic network interdiction. *Operations Research*, 46(2):184–197, 1998. doi: 10.1287/opre.46.2.184.
- [27] Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3):595–612, 2010. doi: 10.1287/opre.1090.0741.
- [28] Oscar Dowson and Lea Kapelevich. SDDP.jl: A Julia package for stochastic dual dynamic programming. *INFORMS Journal on Computing*, 33(1):27–33, 2020. doi: 10.1287/ijoc.2020.0987.
- [29] 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. doi: 10.1287/moor.2020.1085.
- [30] Jitka Dupačová. Optimization under exogenous and endogenous uncertainty. *University of West Bohemia in Pilsen*, 2006.
- [31] Daniel Duque and David P. Morton. Distributionally robust stochastic dual dynamic programming. *SIAM Journal on Optimization*, 30(4):2841–2865, 2020. doi: 10.1137/19M1309602.
- [32] Daniel Duque, Sanjay Mehrotra, and David P. Morton. Distributionally robust two-stage stochastic programming. *SIAM Journal on Optimization*, 32(3):1499–1522, 2022. doi: 10.1137/20M1370227.
- [33] D. R. Fulkerson and Gary C. Harding. Maximizing the minimum source-sink path

- subject to a budget constraint. *Mathematical Programming*, 13(1):116–118, 1977. doi: 10.1007/bf01584329.
- [34] Dinakar Gade, Simge Küçükyavuz, and Suvrajeet Sen. Decomposition algorithms with parametric Gomory cuts for two-stage stochastic integer programs. *Mathematical Programming*, 144(1):39–64, 2014. doi: 10.1007/s10107-012-0615-y.
- [35] Harsha Gangammanavar and Manish Bansal. Stochastic decomposition method for two-stage distributionally robust linear optimization. *SIAM Journal on Optimization*, 32(3):1901–1930, 2022. doi: 10.1137/20M1378600.
- [36] Rui Gao and Anton Kleywegt. Distributionally robust stochastic optimization with Wasserstein distance. *Mathematics of operations research*, 48(2):603–655, 2023. doi: 10.1287/moor.2022.1275.
- [37] Bruce Golden. A problem in network interdiction. *Naval Research Logistics Quarterly*, 25(4):711–713, 1978. doi: 10.1002/nav.3800250412.
- [38] Ralph E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64(5):275–278, 1958. doi: 10.1090/S0002-9904-1958-10224-4.
- [39] Jun-ya Gotoh, Michael Jong Kim, and Andrew EB Lim. A data-driven approach to beating SAA out of sample. *Operations Research*, 73(2):829–841, 2023. doi: 10.1287/opre.2021.0393.
- [40] Harald Held, Raymond Hemmecke, and David L. Woodruff. A decomposition algorithm applied to planning the interdiction of stochastic networks. *Naval Research Logistics*, 52(4):321–328, 2005. doi: 10.1002/nav.20079.

- [41] Lars Hellemo, Paul I. Barton, and Asgeir Tomasgard. Decision-dependent probabilities in stochastic programs with recourse. *Computational Management Science*, 15:369–395, 2018. doi: 10.1007/s10287-018-0330-0.
- [42] Raymond Hemmecke, Rüdiger Schultz, and David L Woodruff. Interdicting stochastic networks with binary interdiction effort. In David L Woodruff, editor, *Network interdiction and stochastic integer programming*, pages 69–84. Kluwer, Boston, MA, 2003.
- [43] Le Thi Khanh Hien, Melvyn Sim, and Huan Xu. Mitigating interdiction risk with fortification. *Operations Research*, 68(2):348–362, 2020. doi: 10.1287/opre.2019.1890.
- [44] Tim Holzmann and J. Cole Smith. The shortest path interdiction problem with randomized interdiction strategies: Complexity and algorithms. *Operations Research*, 69(1):82–99, 2020. doi: 10.1287/opre.2020.2023.
- [45] Eitan Israeli. *System Interdiction and Defense*. PhD thesis, Naval Postgraduate School, Monterey, CA, 1999.
- [46] Eitan Israeli and R. Kevin Wood. Shortest-path network interdiction. *Networks*, 40(2):97–111, 2002. doi: 10.1002/net.10039.
- [47] Udom Janjarassuk and Jeff Linderoth. Reformulation and sampling to solve a stochastic network interdiction problem. *Networks*, 52(3):120–132, 2008. doi: 10.1002/net.20237.
- [48] Nan Jiang and Weijun Xie. Distributionally favorable optimization: A framework for data-driven decision-making with endogenous outliers. *SIAM Journal on Optimization*, 34(1):419–458, 2024. doi: 10.1137/22M1528094.

- [49] Sumin Kang and Manish Bansal. Distributionally risk-receptive and risk-averse network interdiction problems with general ambiguity set. *Networks*, 81(1):3–22, 2023. doi: 10.1002/net.22114.
- [50] Sumin Kang and Manish Bansal. Bi-parameterized two-stage stochastic min-max and min-min mixed integer programs. *arXiv preprint arXiv:2501.01081*, 2025.
- [51] Sumin Kang and Manish Bansal. Distributionally risk-receptive and robust multistage stochastic integer programs and interdiction models. *Mathematical Programming*, 2025. doi: 10.1007/s10107-024-02192-y.
- [52] Kibaek Kim and Sanjay Mehrotra. A two-stage stochastic integer programming approach to integrated staffing and scheduling with application to nurse management. *Operations Research*, 63(6):1431–1451, 2015. doi: 10.1287/opre.2015.1421.
- [53] 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. doi: 10.1007/s10479-022-05087-3.
- [54] Guanghui Lan. Complexity of stochastic dual dynamic programming. *Mathematical Programming*, 191(2):717–754, 2022. doi: 10.1007/s10107-020-01567-1.
- [55] 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. doi: 10.1016/0167-6377(93)90002-X.
- [56] Xiao Lei, Siqian Shen, and Yongjia Song. Stochastic maximum flow interdiction problems under heterogeneous risk preferences. *Computers & Operations Research*, 90: 97–109, 2018. doi: 10.1016/j.cor.2017.09.004.

- [57] Hanyang Li and Ying Cui. A decomposition algorithm for two-stage stochastic programs with nonconvex recourse functions. *SIAM Journal on Optimization*, 34(1):306–335, 2024. doi: 10.1137/22M1488533.
- [58] Junyi Liu, Ying Cui, Jong-Shi Pang, and Suvrajeet Sen. Two-stage stochastic programming with linearly bi-parameterized quadratic recourse. *SIAM Journal on Optimization*, 30(3):2530–2558, 2020. doi: 10.1137/19M1276819.
- [59] 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. doi: 10.1287/mnsc.1030.0164.
- [60] Fengqiao Luo and Sanjay Mehrotra. Distributionally robust optimization with decision dependent ambiguity sets. *Optimization Letters*, 14(8):2565–2594, 2020. doi: 10.1007/s11590-020-01574-3.
- [61] Qi Luo, Viswanath Nagarajan, Alexander Sundt, Yafeng Yin, John Vincent, and Mehrdad Shahabi. Efficient algorithms for stochastic ride-pooling assignment with mixed fleets. *Transportation Science*, 57(4):908–936, 2023.
- [62] 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. doi: 10.1080/0740817X.2011.602659.
- [63] Peyman Mohajerin Esfahani and Daniel Kuhn. Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1):115–166, 2018. doi: 10.1007/s10107-017-1172-1.

- [64] David P. Morton, Feng Pan, and Kevin J. Saeger. Models for nuclear smuggling interdiction. *IIE Transactions*, 39(1):3–14, 2007. doi: 10.1080/07408170500488956.
- [65] Hideaki Nakao. *Distributionally Robust Optimization in Sequential Decision Making*. PhD thesis, University of Michigan, Ann Arbor, 2021.
- [66] 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. doi: 10.1137/19M1268410.
- [67] George Nemhauser and Laurence Wolsey. *Integer and Combinatorial Optimization*. Wiley, 1 edition, 1988. ISBN 978-0-471-82819-8 978-1-118-62737-2.
- [68] Di H. Nguyen and J. Cole Smith. Network interdiction with asymmetric cost uncertainty. *European Journal of Operational Research*, 297(1):239–251, 2022. doi: 10.1016/j.ejor.2021.04.055.
- [69] Di H. Nguyen and J. Cole Smith. Asymmetric stochastic shortest-path interdiction under conditional value-at-risk. *IIE Transactions*, pages 1–32, 2022. doi: 10.1080/24725854.2022.2043570.
- [70] 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.
- [71] Lewis Ntaimo, Julián A. Gallego Arrubla, Curt Stripling, Joshua Young, and Thomas Spencer. A stochastic programming standard response model for wildfire initial attack planning. *Canadian Journal of Forest Research*, 42(6):987–1001, 2012. doi: 10.1139/x2012-032.

- [72] Feng Pan and David P. Morton. Minimizing a stochastic maximum-reliability path. *Networks*, 52(3):111–119, 2008. doi: 10.1002/net.20238.
- [73] 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. doi: 10.1016/j.ejor.2022.06.049.
- [74] Seonghun Park and Manish Bansal. K-submodular interdiction problems under distributional risk-receptiveness and robustness: Application to machine learning. *arXiv Preprint arXiv:2406.13023*, 2025.
- [75] Seonghun Park and Manish Bansal. Algorithms for cameras view-frame placement problems in the presence of an adversary and distributional ambiguity. *IEEE Transactions on Automation Science and Engineering*, 22:353–364, 2025. doi: 10.1109/TASE.2024.3350973.
- [76] Babak Saleck Pay, Jason R. W. Merrick, and Yongjia Song. Stochastic network interdiction with incomplete preference. *Networks*, 73(1):3–22, 2019. doi: 10.1002/net.21831.
- [77] Mario VF Pereira and Leontina MVG Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, 52(1):359–375, 1991. doi: 10.1007/BF01582895.
- [78] Georg Pflug and David Wozabal. Ambiguity in portfolio selection. *Quantitative Finance*, 7(4):435–442, 2007. doi: 10.1080/14697680701455410.
- [79] Andrew B. Philpott and Ziming Guan. On the convergence of stochastic dual dynamic programming and related methods. *Operations Research Letters*, 36(4):450–455, 2008. doi: 10.1016/j.orl.2008.01.013.

- [80] Andrew B. Philpott, Vitor L. de Matos, and Lea Kapelevich. Distributionally robust SDDP. *Computational Management Science*, 15(3):431–454, 2018. doi: 10.1007/s10287-018-0314-0.
- [81] Hamed Rahimian and Sanjay Mehrotra. Frameworks and results in distributionally robust optimization. *Open Journal of Mathematical Optimization*, 3:1–85, 2022. doi: 10.5802/ojmo.15.
- [82] Hamed Rahimian, Güzin Bayraksan, and Tito Homem-de-Mello. Identifying effective scenarios in distributionally robust stochastic programs with total variation distance. *Mathematical Programming*, 173(1):393–430, 2019. doi: 10.1007/s10107-017-1224-6.
- [83] R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *The Journal of Risk*, 2(3):21–41, 2000. doi: 10.21314/JOR.2000.038.
- [84] Johannes O. Royset, Louis L. Chen, and Eric Eckstrand. Rockafellian relaxation and stochastic optimization under perturbations. *Mathematics of Operations Research*, 2024. doi: 10.1287/moor.2022.0122.
- [85] 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. doi: 10.1287/ijoc.2022.1257.
- [86] Javier Salmerón. Deception tactics for network interdiction: A multiobjective approach. *Networks*, 60(1):45–58, 2012. doi: 10.1002/net.20458.
- [87] Javier Salmerón, Kevin Wood, and Ross Baldick. Worst-case interdiction analysis of large-scale electric power grids. *IEEE Transactions on Power Systems*, 24(1):96–104, 2009. doi: 10.1109/TPWRS.2008.2004825.

- [88] Maria P. Scaparra and Richard L. Church. A bilevel mixed-integer program for critical infrastructure protection planning. *Computers & Operations Research*, 35(6):1905–1923, 2008. doi: 10.1016/j.cor.2006.09.019.
- [89] Herbert Scarf. A min-max solution of an inventory problem. In Kenneth Joseph Arrow, Samuel Karlin, and Herbert Scarf, editors, *Studies in the Mathematical Theory of Inventory and Production*, pages 201–209. Stanford University Press, Stanford, CA, 1958.
- [90] Rüdiger Schultz and Stephan Tiedemann. Conditional value-at-risk in stochastic programs with mixed-integer recourse. *Mathematical Programming*, 105:365–386, 2006. doi: 10.1007/s10107-005-0658-4.
- [91] Alexander Shapiro. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, 209(1):63–72, 2011. doi: 10.1016/j.ejor.2010.08.007.
- [92] Hanif D. Sherali and Xiaomei Zhu. Two-stage fleet assignment model considering stochastic passenger demands. *Operations Research*, 56(2):383–399, 2008. doi: 10.1287/opre.1070.0476.
- [93] J. Cole Smith and Yongjia Song. A survey of network interdiction models and algorithms. *European Journal of Operational Research*, 283(3):797–811, 2020. doi: 10.1016/j.ejor.2019.06.024.
- [94] Jun Song and Chaoyue Zhao. Optimistic distributionally robust policy optimization. *arXiv preprint arXiv:2006.07815*, 2020.
- [95] Yongjia Song and Siqian Shen. Risk-averse shortest path interdiction. *INFORMS Journal on Computing*, 28(3):527–539, 2016. doi: 10.1287/ijoc.2016.0699.

- [96] Svyatoslav Trukhanov, Lewis Ntaimo, and Andrew Schaefer. Adaptive multicut aggregation for two-stage stochastic linear programs with recourse. *European Journal of Operational Research*, 206(2):395–406, 2010. doi: 10.1016/j.ejor.2010.02.025.
- [97] Halit Üster and Gökhan Memişoğlu. Biomass logistics network design under price-based supply and yield uncertainty. *Transportation Science*, 52(2):474–492, 2018. doi: 10.1287/trsc.2017.0766.
- [98] Niels van der Laan and Ward Romeijnders. A converging Benders’ decomposition algorithm for two-stage mixed-integer recourse models. *Operations Research*, 72(5):2190–2214, 2023. doi: 10.1287/opre.2021.2223.
- [99] Richard M. Van Slyke and Roger Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM journal on applied mathematics*, 17(4):638–663, 1969. doi: 10.1137/0117061.
- [100] Alan Washburn and Kevin Wood. Two-person zero-sum games for network interdiction. *Operations Research*, 43(2):243–251, 1995. doi: 10.1287/opre.43.2.243.
- [101] Richard Wollmer. Removing arcs from a network. *Operations Research*, 12(6):934–940, 1964. doi: 10.1287/opre.12.6.934.
- [102] R. Kevin Wood. Deterministic network interdiction. *Mathematical and Computer Modelling*, 17(2):1–18, 1993. doi: 10.1016/0895-7177(93)90236-r.
- [103] Jing Yang, Juan S. Borrero, Oleg A. Prokopyev, and Denis Sauré. Sequential shortest path interdiction with incomplete information and limited feedback. *Decision Analysis*, 18(3):218–244, 2021. doi: 10.1287/deca.2021.0426.
- [104] Joyce W. Yen and John R. Birge. A stochastic programming approach to the airline

- crew scheduling problem. *Transportation Science*, 40(1):3–14, 2006. doi: 10.1287/trsc.1050.0138.
- [105] Soovin Yoon, Laura A. Albert, and Veronica M. White. A stochastic programming approach for locating and dispatching two types of ambulances. *Transportation Science*, 55(2):275–296, 2021. doi: 10.1287/trsc.2020.1023.
- [106] 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. doi: 10.1007/s10107-020-01580-4.
- [107] Minjiao Zhang and Simge Küçükyavuz. Finitely convergent decomposition algorithms for two-stage stochastic pure integer programs. *SIAM Journal on Optimization*, 24(4):1933–1951, 2014. doi: 10.1137/13092678X.
- [108] Shixuan Zhang and Xu Andy Sun. Stochastic dual dynamic programming for multistage stochastic mixed-integer nonlinear optimization. *Mathematical Programming*, 196(1):935–985, 2022. doi: 10.1007/s10107-022-01875-8.
- [109] Yingqiu Zhang, Manish Bansal, and Adolfo R. Escobedo. Risk-neutral and risk-averse transmission switching for load shed recovery with uncertain renewable generation and demand. *IET Generation, Transmission & Distribution*, 14(21):4936–4945, 2020. doi: 10.1049/iet-gtd.2020.0964.
- [110] Qipeng P. Zheng, Jianhui Wang, Panos M. Pardalos, and Yongpei Guan. A decomposition approach to the two-stage stochastic unit commitment problem. *Annals of Operations Research*, 210(1):387–410, 2013. doi: 10.1007/s10479-012-1092-7.
- [111] Jikai Zou, Shabbir Ahmed, and Xu Andy Sun. Stochastic dual dynamic integer

programming. *Mathematical Programming*, 175(1):461–502, 2019. doi: 10.1007/s10107-018-1249-5.

Appendices

Chapter A.

Appendix for Chapter 3

A.1 Proofs

A.1.1 Proof of Theorem 3.1

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

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

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

$$D_{max}(\bar{x}) \leq \max_{P \in \mathcal{P}} \sum_{\omega \in \Omega} p_\omega \sum_{a \in A} (c_a + d_a^\omega \zeta_a^\omega \bar{x}_a) \hat{y}_a^\omega, \quad \text{from the above inequalities,} \quad (\text{A.1a})$$

$$= \max_{P \in \mathcal{P}} \sum_{\omega \in \Omega} p_\omega \sum_{a \in A} (c_a + d_a^\omega \zeta_a^\omega \bar{x}_a + d_a^\omega \zeta_a^\omega \hat{x}_a - d_a^\omega \zeta_a^\omega \hat{x}_a) \hat{y}_a^\omega, \quad (\text{A.1b})$$

$$= \max_{P \in \mathcal{P}} \sum_{\omega \in \Omega} p_\omega \sum_{a \in A} \left\{ (c_a + d_a^\omega \zeta_a^\omega \hat{x}_a) \hat{y}_a^\omega + d_a^\omega \zeta_a^\omega \bar{x}_a \hat{y}_a^\omega - d_a^\omega \zeta_a^\omega \hat{x}_a \hat{y}_a^\omega \right\}, \quad (\text{A.1c})$$

$$\leq \max_{P \in \mathcal{P}} \sum_{\omega \in \Omega} p_\omega \sum_{a \in A} (c_a + d_a^\omega \zeta_a^\omega \hat{x}_a) \hat{y}_a^\omega + \max_{P \in \mathcal{P}} \sum_{\omega \in \Omega} p_\omega \sum_{a \in A} d_a^\omega \zeta_a^\omega \hat{y}_a^\omega (\bar{x}_a - \hat{x}_a), \quad (\text{A.1d})$$

$$= D_{max}(\hat{x}) + \max_{P \in \mathcal{P}} \sum_{\omega \in \Omega} p_\omega \sum_{a \in A} d_a^\omega \zeta_a^\omega \hat{y}_a^\omega (\bar{x}_a - \hat{x}_a), \quad (\text{A.1e})$$

$$\leq D_{max}(\hat{x}) + \sum_{a \in A} \max_{P \in \mathcal{P}} \sum_{\omega \in \Omega} p_\omega d_a^\omega \zeta_a^\omega \hat{y}_a^\omega (\bar{x}_a - \hat{x}_a). \quad (\text{A.1f})$$

For $\hat{x} = \bar{x}$, these inequalities hold trivially. Also, in Inequality (A.1e), an optimal solution to the maximization ($\max_{P \in \mathcal{P}}(\cdot)$) depends on \bar{x} , and therefore, it does not provide a linear inequality of the form (3.7). Next, we showcase that

$$\max_{P \in \mathcal{P}} \sum_{\omega \in \Omega} p_\omega d_a^\omega \zeta_a^\omega \hat{y}_a^\omega (\bar{x}_a - \hat{x}_a) = \beta_a(\hat{x}_a) (\bar{x}_a - \hat{x}_a), \quad (\text{A.2})$$

where $\beta_a(\hat{x}_a)$ is given by (3.8). As the interdiction variables are binary, the value of $(\bar{x}_a - \hat{x}_a)$ for each arc $a \in A$ could be one of the three possible values: 0, 1, or -1.

Case I. For $(\bar{x}_a - \hat{x}_a) = 0$, i.e., $\bar{x}_a = \hat{x}_a \in \{0, 1\}$, equation (A.2) holds trivially.

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

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

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

$$P \in \mathcal{P}\}.$$

Therefore, using (A.1f) and (A.2), we conclude that Inequality (3.7) is valid for all $x \in X$. \square

A.1.2 Proof of Theorem 3.2

Proof. In Steps 3-11, the algorithm solves $|\Omega|$ number of subproblems (linear programs), master problem (a mixed-binary program), and a distribution separation problem (3.10) corresponding to the ambiguity set \mathcal{P} . It also derives an optimality cut by calling a distribution separation algorithm at most $|A|$ number of times. Under the assumption of existence of a finitely convergent distribution separation algorithm, Steps 3-11 also takes finite iterations. Therefore, to ensure the finite convergence of Algorithm 3.1, we only need to show that these steps are performed a finite number of times before the algorithm converges to an optimal solution. Given an optimal solution $(\hat{x}^L, \hat{\theta}^L)$ of a master problem, there are two possibilities.

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

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

A.1.3 Proof of Theorem 3.3

Proof. The proof of Theorem 3.3 follows the similar arguments as in the proof of Theorem 3.2, except that an optimality cut is defined by (3.15b), instead of (3.7). \square

A.1.4 Proof of Theorem 3.4

Proof. In Iteration $L \geq 1$ of the relax-and-append algorithm, Algorithm 3.2 (within branch-and-cut framework) solves a relaxation of DRO-SPIP where a distribution separation algorithm is to explicitly enumerate all probability distributions in a finite set $\bar{\mathcal{P}}^L \subset \mathcal{P}$. This requires a finite number of operations due to Theorem 3.3. Also, by the assumption, we can solve an outer distribution separation problem using a finitely convergent oracle. Next, we need to show that the relax-and-append algorithm satisfies the termination condition in finite iterations, i.e., $L < \infty$. Let an optimal solution for the relaxation of DRO-SPIP in iteration L be \bar{x}^L that yields an upper bound $z_{RO}^{ub,L}$. Also, assume that an optimal probability distribution provided by the outer distribution separation problem for $x = \bar{x}^L$ is \bar{P}^L . Now, one of the following two cases can occur: (I) $\bar{x}^L = \bar{x}^{L'}$ for some $L' < L$; or (II) $\bar{x}^L \neq \bar{x}^l$ for all $l < L$. In Case I, since the current relaxed ambiguity set $\bar{\mathcal{P}}^L$ already contains probability distribution $\bar{P}^{L'}$, we get $D_{min}(\bar{x}^L) = D_{min}(\bar{x}^{L'}) = z_{RO}^{ub,L}$. Hence, current solution \bar{x}^L is an optimal solution. In Case II, either of the two subcases can occur: (A) there exists an optimal distribution \bar{P}^L that belongs to $\bar{\mathcal{P}}^L$, which again implies that $D_{min}(\bar{x}^L) = z_{RO}^{ub,L}$ and the current solution is optimal; (B) if there does not exist an optimal distribution \bar{P}^L that belongs to $\bar{\mathcal{P}}^L$, then we proceed to the next iteration with $\bar{\mathcal{P}}^{L+1} = \bar{\mathcal{P}}^L \cup \{\bar{P}^L\}$. Since $|X|$ is finite, there will be only finite iterations until the relax-and-append algorithm leads to Cases I or II(A), thereby returning an optimal solution. \square

A.2 L-Shaped Method with Branch-and-Cut for DRR-SPIP

For the sake of completeness of this chapter, we also provide a discussion on our implementation of the L-shaped method with a branch-and-cut framework to solve reformulation (3.6) of DRR-SPIP, which is equivalent to

$$\max \quad \theta \tag{A.3a}$$

$$\text{s.t.} \quad \theta \leq \sum_{\omega \in \Omega} \sum_{a \in A} (p_\omega c_a + d_a^\omega \zeta_a^\omega \eta_a^\omega) \hat{y}_a^\omega, \quad \text{for all } \hat{y}^\omega \in \hat{Y}_E^\omega, \omega \in \Omega, \tag{A.3b}$$

$$\eta_a^\omega \leq x_a, \quad \forall \omega \in \Omega, a \in A, \tag{A.3c}$$

$$\eta_a^\omega \leq p_\omega, \quad \forall \omega \in \Omega, a \in A, \tag{A.3d}$$

$$\eta_a^\omega \geq 0, \quad \forall \omega \in \Omega, a \in A, \tag{A.3e}$$

$$x \in X, \{p_\omega\}_{\omega \in \Omega} \in \mathcal{P}, \theta \in \mathbb{R}, \tag{A.3f}$$

where \hat{Y}_E^ω is the set of all extreme points of Y^ω for $\omega \in \Omega$. At the root node of the branch-and-cut tree, we construct an upper bound approximation by relaxing constraints (A.3b), which is referred to as current problem. For the boundedness of the root-node problem (in particular, θ), we randomly select a feasible solution $\bar{x}_0 \in X$, solve subproblems (3.4c) with $x = \bar{x}_0$ to obtain $\bar{y}_0^\omega \in \hat{Y}_E^\omega$ for all $\omega \in \Omega$, and derive an optimality cut of the form (A.3b).

Similar to a generic branch-and-cut approach, at each node in the branch-and-cut tree, the algorithm solves the LP relaxation of the current problem (node-LP) with fixed values for a subset of the x variables, i.e., $x_i = 1$ for $i \in S_1$ and $x_i = 0$ for $i \in S_2$ where $S_1, S_2 \subseteq \{1, \dots, |A|\}$ and $S_1 \cap S_2 = \emptyset$. Denote $(\bar{x}, \bar{\theta}, \bar{\eta})$ as the solution to the node-LP, where $\bar{\eta} = \{\bar{\eta}^\omega\}_{\omega \in \Omega, a \in A}$ and $\bar{\theta}$ gives a node upper bound on the optimal objective value of (A.3). Among all leaf nodes in the current tree, we collect the node upper bounds and take the

maximum as the best upper bound (UB). In case \bar{x} is integral, the algorithm computes $R_\omega(\bar{p}_\omega, \bar{\eta}^\omega)$ and associated optimal solution $\bar{y}_{node}^\omega \in \hat{Y}_E^\omega$ for all $\omega \in \Omega$, where $\{\bar{p}_\omega\}$ is obtained by solving $\max_{P \in \mathcal{P}} \sum_{\omega \in \Omega} p_\omega Q(\bar{x})$. If $\bar{\theta} > \sum_{\omega \in \Omega} R_\omega(\bar{p}_\omega, \bar{\eta}^\omega)$, it uses \bar{y}_{node}^ω to derive an optimality cut of the form (A.3b), adds the cut to the current problem (and node-LP), and resolve the node-LP. The foregoing step is repeated until there is no violated optimality cut. Again, in case the latest node-LP solution has integral \bar{x} , the algorithm updates the best-known lower bound, $LB = \max\{LB, \sum_{\omega \in \Omega} R_\omega(\bar{p}_\omega, \bar{\eta}^\omega)\}$, and check if the optimality gap, $(UB - LB)/LB$, is less than a given tolerance level. If the current node-LP solution is not integral, the algorithm proceeds with a conventional branch-and-cut procedure, such as adding cutting planes (and resolving the node-LP), branching, and adding two nodes to the branch-and-bound tree.

Chapter B.

Appendix for Chapter 4

B.1 Formulations of the Network Interdiction Problems

We provide detailed formulations of the multistage NIPs used in our computational tests. We focus on the risk-neutral formulations, as their DRO and DRR variants can be readily derived from these by appropriately maximizing or minimizing with respect to the probability distribution within an ambiguity, as in (4.3) and (4.5).

B.1.1 Formulation of MS-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. At 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 till 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 $c_{t,a}$. 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 (4.1) and (4.2) where, for each $t \in [T]$ and $\omega_t \in \Omega_t$,

$$Q_t(x_{t-1}, \omega_t) := \min \psi_t(x_t, \omega_t) + \mathbb{E}_{P_{t+1}}[Q_{t+1}(x_t, \omega_{t+1})] \quad (\text{B.1a})$$

$$\text{s.t. } x_t \geq x_{t-1} \quad (\text{B.1b})$$

$$\sum_{a \in A} f_{t,a} x_{t,a} \leq b_t + \sum_{a \in A} f_{t,a} x_{t-1,a} \quad (\text{B.1c})$$

$$x_t \in \{0, 1\}^{|A|}, \quad (\text{B.1d})$$

and

$$\psi_t(x_t, \omega_t) := \max_{y_t \geq 0} \left\{ y_{t,(r,s)} : \sum_{a \in \delta^+(n)} y_{t,a} - \sum_{a \in \delta^-(n)} y_{t,a} = 0, \quad \forall n \in N, \quad (\text{B.2a}) \right.$$

$$\left. y_{t,a} \leq c_{t,a}(\omega_t)(1 - x_{t,a}), \quad \forall a \in A \right\}. \quad (\text{B.2b})$$

In interdictor's problem (B.1), constraint (B.1b) ensures that the impact of interdiction on an arc remains till the end of the time horizon, and constraint (B.1c) restricts the total interdiction cost within the given budget b_t . Given an interdiction solution x_t , the objective function of the interdictor's problem at stage $t \in [T]$, $\psi_t(x_t, \omega_t)$, is a value function that provides a maximum flow over the interdicted network. The function $\psi_t(x_t, \omega_t)$ is computed by solving the network user's problem (B.2) where decision variable $y_{t,a}$ represents a flow on

arc a in A , constraints (B.2a) enforce the flow balance on nodes in N , and constraints (B.2b) restrict the capacity of arcs in A . Notice that because of constraints (B.2b), $y_{t,a}$ is restricted to be zero if the interdiction occurs on arc $a \in A$, i.e., if $x_{t,a} = 1$.

To solve each stage's problem during our tests, we first dualize the network user's problem (B.2) and then reformulate the problem (B.1) into a single-level minimization mixed-binary program.

B.1.2 Formulation of MS-FLIP

The single-stage (deterministic) FLIP, also known as the r -interdiction median problem, is introduced by Church et al. [24]. The objective of the single-stage FLIP is to find a subset of r facilities whose removal maximizes the network user's objective of minimizing the total weighted distance. MS-FLIP extends this problem to a multistage stochastic setting, where interdiction decisions are made sequentially over multiple time periods under demand uncertainty.

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 $y_{ilm} \in \{0, 1\}$ 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 $c_{ilm}(\omega_t) = a_{il}(\omega_t)d_{lm}$, where $d_{lm} > 0$ is the Euclidean distance between l and m , and $a_{il}(\omega_t)$ is an uncertain demand at point $l \in [L]$,

for each scenario $\omega_t \in \Omega_t$. The formulation of MS-FLIP is given by (4.1) and (4.2) with

$$Q_t(x_{t-1}, \omega_t) := \max \sum_{l \in [L], m \in [M]} c_{tlm}(\omega_t) y_{tlm} + \mathbb{E}_{P_{t+1}} [Q_{t+1}(x_t, \omega_{t+1})] \quad (\text{B.3a})$$

$$\text{s.t. } \sum_{m \in [M]} y_{tlm} = 1, \quad \forall l \in [L], \quad (\text{B.3b})$$

$$x_t \geq x_{t-1}, \quad (\text{B.3c})$$

$$\sum_{m \in [M]} (x_{tm} - x_{t-1,m}) = r_t, \quad (\text{B.3d})$$

$$\sum_{n \in S_{lm}} y_{tln} \leq x_{tm}, \quad \forall l \in [L], m \in [M], \quad (\text{B.3e})$$

$$x_t \in \{0, 1\}^M, \quad y_t \in \{0, 1\}^{L \times M} \quad (\text{B.3f})$$

for each $\omega_t \in \Omega_t$. 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 (B.3a) represents the total weighted distance of the assignment of demand points to non-interdicted facilities. Constraints (B.3b) enforce each demand point to be assigned to a facility. Constraint (B.3c) ensures that the facilities interdicted from the previous stages remain interdicted for the current stage. Constraint (B.3d) ensures that the total number of interdictions occurred at the current stage equals to the budget r_t . Constraint (B.3e), 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 (B.3) is a single-level maximization problem, but not the max-min form of typical interdiction problems, 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 (B.3e).

B.2 Proofs

B.2.1 Proof of Theorem 4.1

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

$$\begin{aligned}
\min_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i \hat{Q}_{t+1}^l(x_t, \omega_{t+1}^i) &\geq \min_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i \left((\alpha_t^{i,k})^\top x_t + \beta_t^{i,k} \right) \\
&= \min_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i \left((\alpha_t^{i,k})^\top (x_t - \hat{x}_t) + (\alpha_t^{i,k})^\top \hat{x}_t + \beta_t^{i,k} \right) \\
&\geq \gamma_t^k + \min_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i (\alpha_t^{i,k})^\top (x_t - x_t^k) \\
&\geq \gamma_t^k + \sum_{j \in [d_x]} \min_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i \alpha_{t,j}^{i,k} (x_{t,j} - x_{t,j}^k).
\end{aligned}$$

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

$$\pi_{t,j}^k = \min_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i \alpha_{t,j}^{i,k}.$$

If $x_{t,j}^k = 1$, then $(x_{t,j} - x_{t,j}^k) \in \{0, -1\}$ and we can fix the coefficient to

$$\pi_{t,j}^k = \max_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i \alpha_{t,j}^{i,k}.$$

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

$$\min_{P_{t+1} \in \mathcal{P}_{t+1}} \sum_{i \in \mathcal{N}_{t+1}} p_{t+1}^i \hat{Q}_{t+1}^l(x_t, \omega_{t+1}^i) \geq (\pi_t^k)^\top (x_t - x_t^k) + \gamma_t^k, \quad \forall k \in [K_t^l]. \quad (\text{B.4})$$

Function $\phi_t^{l,C}(x_t)$ is constructed using the affine functions in the right-hand side of (B.4), and thus it follows that $\phi_t^{l,C}(x_t) \leq \mathcal{Q}_{t+1}^{RR}(x_t)$. \square

B.2.2 Proof of Proposition 4.2

Proof. Function $\mathcal{Q}_t^{RR}(x_{t-1}) = -\max_{P_t \in \mathcal{P}_t(x_{t-1})} \mathbb{E}_{P_t}[-Q_t^{RR}(x_{t-1}, \omega_t)]$. By applying Theorem 1 in [36] to the maximization problem, the dual formulation is given by

$$\begin{aligned} \mathcal{Q}_t^{RR}(x_{t-1}) &= -\min_{\rho_t \geq 0, v_t^i} \epsilon_t(x_{t-1})\rho_t + \sum_{i \in \mathcal{N}_t} \bar{p}_t^i v_t^i \\ &\text{s.t. } \rho_t \|\omega_t - \omega_t^i\| + v_t^i \geq -Q_t^{RR}(x_{t-1}, \omega_t), \quad \forall \omega_t \in \Omega_t, i \in \mathcal{N}_t. \end{aligned}$$

where $\bar{p}_t^i = 1/N_t$. Consequently, we obtain

$$\begin{aligned} &\mathcal{Q}_t^{RR}(x_{t-1}) \\ &= -\min_{\rho_t \geq 0} \left\{ \epsilon_t(x_{t-1})\rho_t + \sum_{i \in \mathcal{N}_t} \bar{p}_t^i \max_{\omega_t \in \Omega_t} \left\{ -\rho_t \|\omega_t - \omega_t^i\| - Q_t^{RR}(x_{t-1}, \omega_t) \right\} \right\} \\ &= \max_{\rho_t \geq 0} \left\{ -\epsilon_t(x_{t-1})\rho_t + \sum_{i \in \mathcal{N}_t} \bar{p}_t^i \min_{\omega_t \in \Omega_t} \left\{ \rho_t \|\omega_t - \omega_t^i\| + Q_t^{RR}(x_{t-1}, \omega_t) \right\} \right\}. \end{aligned}$$

\square

B.2.3 Proof of Theorem 4.3

Proof. By Proposition 4.2, we have

$$\mathcal{Q}_t^{RR}(x_{t-1}) = \max_{\rho_t \geq 0} \left\{ -\epsilon_t(x_{t-1})\rho_t + \sum_{i \in \mathcal{N}_t} \bar{p}_t^i V_t^i(x_{t-1}, \rho_t) \right\}, \quad (\text{B.5})$$

where $\bar{p}_t^i = 1/N_t$ and

$$V_t^i(x_{t-1}, \rho_t) := \min_{\omega_t \in \Omega_t} \left\{ \rho_t \|\omega_t - \omega_t^i\| + Q_t^{RR}(x_{t-1}, \omega_t) \right\} \text{ for } i \in \mathcal{N}_t. \quad (\text{B.6})$$

Using the valid cuts with coefficients $\{(\pi_t^k, \gamma_t^k)\}_{k \in [K_t]}$ for $Q_t^{RR}(x_t)$, we obtain a lower-bounding approximation \underline{V}_t^i of V_t^i under assumptions 4.3 and 4.5, where

$$\underline{V}_t^i(x_{t-1}, \rho_t) := \min_{\omega_t, x_t} \rho_t \|\omega_t - \omega_t^i\| + c_t^\top x_t + \theta_t \quad (\text{B.7a})$$

$$\text{s.t. } A_t x_t - \omega_t \geq -C_t x_{t-1} \quad (\text{B.7b})$$

$$\omega_t \in \Omega_t = [l_t, u_t]^{d_\omega} \quad (\text{B.7c})$$

$$\theta_t \geq (\pi_t^k)^\top x_t + \gamma_t^k, \quad \forall k \in [K_t] \quad (\text{B.7d})$$

$$x_t \in \mathcal{X}_t, \omega_t \in \mathbb{R}^{d_\omega}, \theta_t \in \mathbb{R}. \quad (\text{B.7e})$$

Then, a Lagrangian relaxation of (B.7) without integrality restrictions on x_t , where λ_t^i , (μ_t^i, ν_t^i) , ζ_{tk}^i are dual multipliers associated with constraints (B.7b), (B.7c), (B.7d), respectively, is given by

$$\begin{aligned} \min_{(x_t, \omega_t, \theta_t) \in \mathbb{R}^{d_x + d_\omega + 1}} & -(\lambda_t^i)^\top C_t x_{t-1} + (\mu_t^i)^\top l_t - (\nu_t^i)^\top u_t + \rho_t \|\omega_t - \omega_t^i\| \\ & + (\lambda_t^i - \mu_t^i + \nu_t^i)^\top \omega_t + \left(c_t - \lambda_t^i A_t + \sum_{k \in [K_t]} \pi_t^k \right)^\top x_t \\ & + \left(1 - \sum_{k \in [K_t]} \zeta_{tk}^i \right) \theta_t + \sum_{k \in [K_t]} \gamma_t^k \zeta_{tk}^i. \end{aligned} \quad (\text{B.8})$$

Next, we prove that the Lagrangian dual is equivalent to

$$\max - (\lambda_t^i)^\top C_t x_{t-1} + (\mu_t^i)^\top l_t - (\nu_t^i)^\top u_t + (\lambda_t^i - \mu_t^i + \nu_t^i)^\top \omega_t^i + \sum_{k \in [K_t]} \gamma_t^k \zeta_{tk} \quad (\text{B.9a})$$

$$\text{s.t. } A_t^\top \lambda_t^i - \sum_{k \in [K_t]} \pi_t^k \zeta_{tk} = c_t \quad (\text{B.9b})$$

$$\left\| \lambda_t^i - \mu_t^i + \nu_t^i \right\|_* \leq \rho_t \quad (\text{B.9c})$$

$$\sum_{k \in [K_t]} \zeta_{tk}^i = 1 \quad (\text{B.9d})$$

$$(\lambda_t^i, \mu_t^i, \nu_t^i, \zeta_t^i) \geq 0. \quad (\text{B.9e})$$

As a consequence, we can substitute V_t^i for all scenarios $i \in \mathcal{N}_t$ in (B.5) with these Lagrangian dual and obtain problem (4.20). Since the Lagrangian duals are maximization problems, any feasible solution of them yields a valid cut in the form of (4.18), and tightest cuts are attained by optimal solutions.

We now prove that (B.9) is a Lagrangian dual. Constraints (B.9b) and (B.9d) are straightforwardly derived to ensure $(c_t - \lambda A_t + \sum_{k \in [K_t]} \pi_t^k) = 0$, and $(1 - \sum_{k \in [K_t]} \zeta_k) = 0$, thereby ensuring that (B.8) is not unbounded because of unrestricted x_t and θ_t . Likewise, as shown below, constraint (B.9c) ensures that

$$\tau := \inf_{\omega_t \in \mathbb{R}^{d_\omega}} \{ \rho_t \left\| \omega_t - \omega_t^i \right\| + (\lambda_t^i - \mu_t^i + \nu_t^i)^\top \omega_t \} > -\infty.$$

For the ease of exposition, we use (λ, μ, ν) instead of $(\lambda_t^i, \mu_t^i, \nu_t^i)$ in the following claim.

Claim B.1. *Let $\|\cdot\|_*$ be the dual norm of $\|\cdot\|$. Given $\rho_t \geq 0$, we have $\tau > -\infty$ if and only if $(\rho_t - \|\lambda - \mu + \nu\|_*) \geq 0$. Also, if $(\rho_t - \|\lambda - \mu + \nu\|_*) \geq 0$, then $\tau = (\lambda - \mu + \nu)^\top \omega_t^i$.*

Proof. By letting $\omega'_t = \omega_t^i - \omega_t$, the infimum τ can be rewritten as follows.

$$\tau = (\lambda - \mu + \nu)^\top \omega_t^i + \inf_{\omega'_t \in \mathbb{R}^{d_\omega}} \left\{ \rho_t \|\omega'_t\| - (\lambda - \mu + \nu)^\top \omega'_t \right\}. \quad (\text{B.10a})$$

Let $\tau^0 := \inf_{\omega'_t \in \mathbb{R}^{d_\omega}} \left\{ \rho_t \|\omega'_t\| - (\lambda - \mu + \nu)^\top \omega'_t \right\}$. If $(\rho_t - \|\lambda - \mu + \nu\|_*) \geq 0$, then we have

$$\tau^0 \geq \inf_{\omega'_t \in \mathbb{R}^{d_\omega}} \left\{ \|\omega'_t\| \cdot (\rho_t - \|\lambda - \mu + \nu\|_*) \right\} = 0.$$

The inequality holds by Hölder's inequality. The infimum $\tau^0 = 0$ since $\omega'_t = 0$ is feasible.

Therefore, $\tau = (\lambda - \mu + \nu)^\top \omega_t^i$.

Now we show the “only-if” direction using proof by contradiction. Suppose that $\tau > -\infty$ and $(\rho_t - \|\lambda - \mu + \nu\|_*) < 0$. This implies that τ^0 is also bounded. By definition of dual norm, i.e., $\|\lambda - \mu + \nu\|_* = \max_{\omega_t \in \mathbb{R}^{d_\omega}} \{\omega_t^\top (\lambda - \mu + \nu) : \|\omega_t\| \leq 1\}$, there exists $\bar{\omega}_t \in \mathbb{R}^{d_\omega}$ such that $\rho_t < \bar{\omega}_t^\top (\lambda - \mu + \nu)$ and $\|\bar{\omega}_t\| \leq 1$. Thus, $\rho_t \|\bar{\omega}_t\| - \bar{\omega}_t^\top (\lambda - \mu + \nu) < 0$. Considering scalar $r \rightarrow \infty$, we have $\rho_t \|r\bar{\omega}_t\| - (r\bar{\omega}_t)^\top (\lambda - \mu + \nu) \rightarrow -\infty$. By taking $\omega'_t = r\bar{\omega}_t$, it follows that $\tau^0 = -\infty$, and $\tau = -\infty$. □

□

B.2.4 Proof of Theorem 4.4

Proof. We first show that the finiteness of the while loop. Let $\{\bar{x}_t^l(\omega_{[t]}), \bar{y}_t^l(\omega_{[t]}) : t \in [T]\}$ 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 $\phi_t^{l,C}(\bar{x}_t^l(\omega_{[t]})) = \mathcal{Q}_{t+1}^{RR}(\bar{x}_t^l(\omega_{[t]}))$ for all $t \in [T-1]$ and all $\omega_{[T]} \in \Omega$.

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 $\phi_t^{l,C}(\bar{x}_t^l(\omega_{[t]})) < \mathcal{Q}_{t+1}^{RR}(\bar{x}_t^l(\omega_{[t]}))$ for some $\omega_{[T]} \in \Omega$. Also, given iteration m , let $\bar{X}_t^m, t \in [T-1]$, be the set of $\bar{x}_t^l(\omega_{[t]})$ for all future iterations $l \in \mathcal{L} \setminus [m]$ and $\omega_{[T]} \in \Omega$, i.e., $\bar{X}_t^m = \{\bar{x}_t^l(\omega_{[t]}) : \omega_{[T]} \in \Omega, l > m, l \in \mathcal{L}\}$.

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

$$\phi_{T-1}^{m,C}(\bar{x}) \geq \min_{P_T \in \mathcal{P}_T} \sum_{i \in \mathcal{N}_T} p_T^i \hat{Q}_T^l(\bar{x}, \omega_T^i) = \mathcal{Q}_T^{RR}(\bar{x}). \quad (\text{B.11})$$

The above holds since the cut added at iteration l (in the form of Constraint (4.10)) satisfies both (4.26a) and (4.26b), and by definition \hat{Q}_T^l is equivalent to Q_T^{RR} . It follows that $\phi_{T-1}^{m,C}(\bar{x}) = \mathcal{Q}_T^{RR}(\bar{x})$ for any $m > l$. Since $|\bar{X}_{T-1}^l| < \infty$ and every $\omega_{[T]} \in \Omega$ has a positive probability to be sampled, it holds with probability one that $\phi_{T-1}^{l,C}(\bar{x}) = \mathcal{Q}_T^{RR}(\bar{x})$ for all $\bar{x} \in \bar{X}_{T-1}^l$ 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 $\phi_{T-2}^{l,C}(\bar{x}) < \mathcal{Q}_{T-1}^{RR}(\bar{x})$ for any $\bar{x} \in \bar{X}_{T-2}^{l_{T-1}}$. Then, we have $\phi_{T-2}^{m,C}(\bar{x}) = \mathcal{Q}_{T-1}^{RR}(\bar{x})$ for all future iterations $m > l$ since

$$\begin{aligned} \phi_{T-2}^{m,C}(\bar{x}) &\geq \min_{P_{T-1} \in \mathcal{P}_{T-1}} \sum_{i \in \mathcal{N}_{T-1}} p_{T-1}^i \hat{Q}_{T-1}^l(\bar{x}, \omega_{T-1}^i) \\ &= \min_{P_{T-1} \in \mathcal{P}_{T-1}} \sum_{i \in \mathcal{N}_{T-1}} p_{T-1}^i \mathcal{Q}_{T-1}^{RR}(\bar{x}, \omega_{T-1}^i) = \mathcal{Q}_{T-1}^{RR}(\bar{x}). \end{aligned} \quad (\text{B.12})$$

The first equality holds since $\phi_{T-1}^{m,C}(\bar{x}) = \mathcal{Q}_T^{RR}(\bar{x})$ for $m > l_{T-1}$. So, we have $\phi_{T-2}^{m,C}(\bar{x}) = \mathcal{Q}_{T-1}^{RR}(\bar{x})$ for any future iteration $m > l$. Since $|\bar{X}_{T-2}^{l_{T-1}}| < \infty$ and every $\omega_{[T]} \in \Omega$ has a positive probability, there exists a finite iteration l such that $\phi_{T-2}^{l,C}(\bar{x}) = \mathcal{Q}_{T-1}^{RR}(\bar{x})$ for all $\bar{x} \in \bar{X}_{T-2}^{l_{T-1}}$ 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 10 is executed in a finite time by assumption. \square

B.2.5 Proof of Theorem 4.5

Proof. By following a similar argument to the proof of Theorem 4.4, 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. \square

B.2.6 Proof of Theorem 4.6

Proof. Let $\{\bar{x}_t^l(\omega_{[t]}), \bar{y}_t^l(\omega_{[t]})\}_{t \in [T]}$ be a policy that is defined by the forward step at iteration l . To show its optimality, it suffices to show $\phi_t^{l,S}(\bar{x}_t^l(\omega_{[t]})) = \mathcal{Q}_{t+1}^{RA}(\bar{x}_t^l(\omega_{[t]}))$ for $t \in [T-1]$ and $\omega_{[T]} \in \Omega$. We can prove this by following a similar argument to the proof of Theorem 4.4. \square

B.2.7 Proof of Proposition 4.7

Proof. 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

$$\mathcal{F}_t^l(\omega_t) := \left\{ (x_{t-1}, z_t) \in \mathbb{R}_+^{d_x} \times \mathbb{R}_+^{d_x+d_y+1} : \right. \\ \left. \bigvee_{h \in H_t} \left(\phi_t - (\pi_t^k)^\top x_t \geq \gamma_t^k, k \in [K_t^l], \right. \right. \\ \left. \left. A_t^h(\omega_t)x_t + B_t^h(\omega_t)y_t + C_t^h(\omega_t)x_{t-1} \geq b_t^h(\omega_t) \right) \right\}, \quad (\text{B.13})$$

and $\mathcal{E}(\bar{x}_{t-1}) := \{(x_{t-1}, z_t) : x_{t-1} = \bar{x}_{t-1}\}$. We claim that $\text{conv}(\mathcal{F}_t^l(\omega_t) \cap \mathcal{E}(\bar{x}_{t-1})) = \text{conv}(\mathcal{F}_t^l(\omega_t)) \cap \mathcal{E}(\bar{x}_{t-1})$. Clearly, $\text{conv}(\mathcal{F}_t^l(\omega_t) \cap \mathcal{E}(\bar{x}_{t-1})) \subseteq \text{conv}(\mathcal{F}_t^l(\omega_t)) \cap \mathcal{E}(\bar{x}_{t-1})$.

To show that $\text{conv}(\mathcal{F}_t^l(\omega_t) \cap \mathcal{E}(\bar{x}_{t-1})) \supseteq \text{conv}(\mathcal{F}_t^l(\omega_t)) \cap \mathcal{E}(\bar{x}_{t-1})$, pick any point $(\bar{x}_{t-1}, \bar{z}_t) \in \text{conv}(\mathcal{F}_t^l(\omega_t)) \cap \mathcal{E}(\bar{x}_{t-1})$. Then, there exist $(x_{t-1}^j, z_t^j) \in \mathcal{F}_t^l(\omega_t)$ and $\lambda^j \in (0, 1], j = 1, \dots, J$ 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, $(x_{t-1}^j, z_t^j) \in \mathcal{F}_t^l(\omega_t) \cap \mathcal{E}(\bar{x}_{t-1}), j \in [J]$, and $(\bar{x}_{t-1}, \bar{z}_t) \in \text{conv}(\mathcal{F}_t^l(\omega_t) \cap \mathcal{E}(\bar{x}_{t-1}))$. This completes the proof of the claim.

To obtain $\text{conv}(\mathcal{F}_t^l(\omega_t))$, we first use Theorem 2.1 in [2] and derive a tight extended formu-

lation of $\mathcal{F}_t^l(\omega_t)$, which is given by

$$\begin{aligned} \widehat{\mathcal{F}}_t^l(\omega_t) := & \left\{ \sum_{h \in H_t} \zeta_{t,0}^h = 1, \sum_{h \in H_t} \zeta_{t,1}^h - x_t = 0, \sum_{h \in H_t} \zeta_{t,2}^h - y_t = 0, \right. \\ & \sum_{h \in H_t} \zeta_{t,3}^h - x_{t-1} = 0, \sum_{h \in H_t} \zeta_{t,4}^h - \phi_t = 0, \\ & A_t^h(\omega_t)\zeta_{t,1}^h + B_t^h(\omega_t)\zeta_{t,2}^h + C_t^h(\omega_t)\zeta_{t,3}^h - b_t^h(\omega_t)\zeta_{t,0}^h \geq 0, \quad h \in H_t, \\ & \zeta_{t,4}^h - (\pi_t^k)^\top \zeta_{t,1}^h - \gamma_t^k \zeta_{t,0}^h \geq 0, \quad h \in H_t, k \in K_t^l, \\ & x_t \in \mathbb{R}_+^{d_x}, y_t \in \mathbb{R}_+^{d_y}, x_{t-1} \in \mathbb{R}_+^{d_x}, \phi_t \in \mathbb{R}_+, \\ & \left. \zeta_{t,0}^h \in \mathbb{R}_+, \zeta_{t,1}^h \in \mathbb{R}_+^{d_x}, \zeta_{t,2}^h \in \mathbb{R}_+^{d_y}, \zeta_{t,3}^h \in \mathbb{R}_+^{d_x}, \zeta_{t,4}^h \in \mathbb{R}_+, h \in H_t \right\}. \end{aligned} \quad (\text{B.14})$$

Since $\mathcal{F}_t^l(\omega_t)$ is unbounded, the projection of the above formulation (B.14) 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 (B.13). They are nonempty and have identical recession cones. This implies that $\text{conv}(\mathcal{F}_t^l(\omega_t))$ is a polyhedron, and thus the closed convex hull of $\mathcal{F}_t^l(\omega_t)$ is equivalent to $\text{conv}(\mathcal{F}_t^l(\omega_t))$. Hence, using the tight extended formulation (B.14), the convex hull of $\mathcal{D}_t^l(\bar{x}_{t-1}, \omega_t)$ is given by $\text{Proj}_{x_{t-1}, z_t}(\widehat{\mathcal{F}}_t^l(\omega_t)) \cap \mathcal{E}(\bar{x}_{t-1})$, which is equivalent to the projection of the set $\widetilde{\mathcal{D}}_t^l(\bar{x}_{t-1}, \omega_t)$ onto the z_t -space. \square

B.2.8 Proof of Proposition 4.8

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

$$\bigvee_{(J_1, J_2) \in \mathcal{J}_t^s} \left(x_{t,j} = 0, j \in J_1, x_{t,j} = 1, j \in J_2, \phi_t - (\pi_t^k)^\top x_t \geq \gamma_t^k, k \in [K_t^l], \right. \\ \left. A_t(\omega_t)x_t + B_t(\omega_t)y_t \geq b_t(\omega_t) - C_t(\omega_t)x_{t-1} \right), \quad (\text{B.15})$$

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 4.7 to this disjunctive set, we obtain the tight extended formulation (4.31). \square

B.3 Pseudocode of DA-SDDP-DP Algorithm

Algorithm B.1: DA-SDDP-DP

```

Initialize:  $l \leftarrow 1$ ;  $x_0 \leftarrow$  initial state;  $\omega_1 \leftarrow$  data at the first stage;
               $\Omega_1 := \{\omega_1\}$ ;  $K_t^l \leftarrow 0$  for  $t = 1, \dots, T - 1$ ;
1 while (satisfying none of stopping conditions) do
2   Sample a scenario path  $\omega^{(l)} \in \Omega := \Omega_1 \times \dots \times \Omega_T$ ;
   // Forward Step
3   for  $t \in [T]$  do
4     Solve  $t$ -stage LP-subproblem (4.27) given  $x_{t-1} = x_{t-1}^l$  and  $\omega_t = \omega_t^{(l)}$ ;
5   end
   // Backward Step
6   for  $t = T, \dots, 2$  do
7     for  $i \in \mathcal{N}_t$  do
8       Solve  $t$ -stage LP-subproblem (4.27) given  $x_{t-1} = x_{t-1}^l$  and  $\omega_t = \omega_t^i$  and
          obtain cut  $(\sigma_{t-1,1}^{i,l}, \sigma_{t-1,0}^{i,l})$ ;
9     end
10    Add cuts  $(\pi_{t-1}^l, \gamma_{t-1}^l)$  to  $(t-1)$ -stage LP-subproblem (4.27) by using cuts
         $(\sigma_{t-1,1}^{i,l}, \sigma_{t-1,0}^{i,l}), i \in \mathcal{N}_t$ ;
11     $K_{t-1}^l \leftarrow K_{t-1}^l + 1$ ;
12  end
13  Solve LP-subproblem (4.27) for  $t = 1$  to obtain the bound  $LB$ ;
14   $K_t^{l+1} \leftarrow K_t^l$  for  $t = 1, \dots, T - 1$ ;
15   $l \leftarrow l + 1$ ;
16 end
17 Return: LP-subproblems,  $LB$ 

```

In this algorithm, let x_t^l be an optimal solution obtained by solving the LP-subproblem (4.29) during a forward step (Line 4) for iteration l and stage t . In a backward step (Line 8), the algorithm solves the LP-subproblem (4.29) given x_t^l and ω_t^i to obtain a cut in the form of Benders cut, $(\sigma_{t-1,1}^{i,l}, \sigma_{t-1,0}^{i,l})$, where $\sigma_{t-1,1}^{i,l}$ and $\sigma_{t-1,0}^{i,l}$ are the optimal dual multipliers,

associated with the constraints $\sum_{h \in H_t} \zeta_{t,0}^h = 1$ and $\sum_{h \in H_t} \zeta_{t,3}^h = x_{t-1}^l$, respectively. These cuts $(\sigma_{t-1,1}^{i,l}, \sigma_{t-1,0}^{i,l})$ for $i \in \mathcal{N}_t$ are used to derive a cut $(\pi_{t-1}^l, \gamma_{t-1}^l)$; the cut $(\pi_{t-1}^l, \gamma_{t-1}^l)$ takes the form of (4.10), if we solve a DRR-MSDP, and it takes the form of (4.22), if we solve a DRO-MSDP. Then, its copies for $h \in H_t$ are added to the $(t-1)$ -stage LP-subproblem (Line 10). In Line 13, 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 DRO-SDDP-DP algorithms, defined as Algorithm B.1 with cut $(\pi_{t-1}^l, \gamma_{t-1}^l)$ obtained as above for DRR-MSDPs and DRO-MSDPs, respectively, have the finite convergence that can be proved using Theorems 4.4 and 4.6, respectively.

Remark B.1. In the implementation of Algorithm B.1, we can establish the LP-subproblems once and reuse them in each iteration, without the need for repeated construction, by adding constraints $\zeta_{t,4}^h - (\pi_t^k)^\top \zeta_{t,1}^h - \gamma_t^k \zeta_{t,0}^h$ to the LP-subproblems as needed (in Line 10).

Chapter C.

Appendix for Chapter 5

C.1 Proofs

C.1.1 Proof of Theorem 5.1

Proof. Clearly, $Q(x, \omega) \leq D(x, \omega)$ for any $\omega \in \Omega$ and $x \in X$. Therefore, to prove the statement, it suffices to show that $Q(x, \omega) \geq D(x, \omega)$, for all $\omega \in \Omega$ and $x \in X$. Fix ω , and let $Z(\omega)$ denote the feasible set of the Lagrangian relaxation (5.5). Also, we let $\tilde{T}(\omega) \in \mathbb{Q}^{\tilde{m} \times n_1}$, $\tilde{W}(\omega) \in \mathbb{Q}^{\tilde{m} \times n_2}$, and $\tilde{r}(\omega) \in \mathbb{Q}^{\tilde{m}}$ such that $\text{conv}(Z(\omega)) = \{(y, z) \in \mathbb{R}_+^{n_2} \times \mathbb{R}_+^{n_1} : \tilde{T}(\omega)z + \tilde{W}(\omega)y \leq \tilde{r}(\omega)\}$. Then, we have

$$\begin{aligned} L(x, \lambda(\omega), \omega) &= \max_{y \in \mathbb{R}_+^{n_2}, z \in \mathbb{R}_+^{n_1}} \left\{ q(\omega)^\top y + x^\top G(\omega)y + \lambda(\omega)^\top z : \tilde{T}(\omega)z + \tilde{W}(\omega)y \leq \tilde{r}(\omega) \right\} \\ &= \min_{\pi \in \mathbb{R}_+^{\tilde{m}}} \left\{ \tilde{r}(\omega)^\top \pi : \tilde{T}(\omega)^\top \pi \geq \lambda(\omega), \tilde{W}(\omega)^\top \pi \geq q(\omega) + G(\omega)^\top x \right\}, \end{aligned}$$

where π represents dual variables. By incorporating the above into the Lagrangian dual (5.6), we have

$$D(x, \omega) = \min_{\lambda(\omega) \in \mathbb{R}^{n_1}, \pi \in \mathbb{R}_+^{\tilde{m}}} \left\{ \tilde{r}(\omega)^\top \pi - \lambda(\omega)^\top x : \tilde{T}(\omega)^\top \pi - \lambda(\omega) \geq 0, \tilde{W}(\omega)^\top \pi \geq q(\omega) + G(\omega)^\top x \right\}. \quad (\text{C.1})$$

Let $z \in \mathbb{R}_+^{n_1}$ and $y \in \mathbb{R}_+^{n_2}$ denote dual variables for the constraints. Then, the resulting dual formulation is

$$\max_{y \in \mathbb{R}_+^{n_2}, z \in \mathbb{R}_+^{n_1}} \left\{ q(\omega)^\top y + x^\top G(\omega)y : \tilde{T}(\omega)z + \tilde{W}(\omega)y \leq \tilde{r}(\omega), z = x \right\}. \quad (\text{C.2})$$

The feasible region of (C.2) can be seen as the intersection of $\text{conv}(Z(\omega))$ and $\{(y, z) : z = x\}$. By Assumption 5.2, $Z(\omega)$ is independent of z_i for $i \in [n_1] \setminus I$; thus, this intersection can be equivalently written as $\text{conv}(Z(\omega)) \cap \{(y, z) : z_i = x_i, \forall i \in I\}$. Since z_i is a binary variable for $i \in I$, the intersection defines a face of $\text{conv}(Z(\omega))$ for any $x \in X$. By the properties of faces, the y component of any extreme point in this set satisfies the integrality constraints, i.e., $y \in \mathcal{Y}$. Consequently, the y component of any optimal solution to (C.2) is feasible to the recourse problem (5.4), which implies that $D(x, \omega) \leq Q(x, \omega)$. \square

C.1.2 Proof of Lemma 5.2

Proof. Fix $(\hat{x}, \hat{\lambda}(\omega), \omega) \in X \times \mathbb{R}_+^{n_1} \times \Omega$ and let $(\hat{y}(\omega), \hat{z}(\omega))$ be an optimal solution of the Lagrangian relaxation (5.5). There can be two cases: (a) $\hat{z}_i(\omega) = \hat{x}_i$ for all $i \in I$ or (b) $\hat{z}_i(\omega) \neq \hat{x}_i$ for some $i \in I$. Clearly, if Case (a), then $\hat{y}(\omega)$ is feasible to the recourse problem (5.2), and thus $q(\omega)^\top \hat{y}(\omega) + \hat{x}^\top G(\omega)\hat{y}(\omega) \leq Q(\hat{x}, \omega)$. Here, the equality holds since we have $Q(\hat{x}, \omega) \leq L(\hat{x}, \hat{\lambda}(\omega), \omega) - \hat{\lambda}(\omega)^\top \hat{x} = q(\omega)^\top \hat{y}(\omega) + \hat{x}^\top G(\omega)\hat{y}(\omega)$. This implies that the solution $\hat{y}(\omega)$ is optimal to the recourse problem, and $\hat{\lambda}(\omega)$ is an optimal solution to the Lagrangian dual.

Now, consider Case (b). Pick any $\bar{y} \in Y(\hat{x}, \omega)$. Then, since solution $(y, z) = (\bar{y}, \hat{x})$ is non-optimal and feasible to the Lagrangian relaxation (5.5), we have

$$q(\omega)^\top \hat{y}(\omega) + \hat{x}^\top G(\omega)\hat{y}(\omega) + \hat{\lambda}(\omega)^\top \hat{z}(\omega) > q(\omega)^\top \bar{y} + \hat{x}^\top G(\omega)\bar{y} + \hat{\lambda}(\omega)^\top \hat{x}. \quad (\text{C.3})$$

By rearranging the terms, we obtain

$$(q(\omega) + G(\omega)^\top \hat{x})^\top (\hat{y}(\omega) - \bar{y}) > \hat{\lambda}(\omega)^\top (\hat{x} - \hat{z}(\omega)) = \sum_{i \in I: \hat{z}_i(\omega) \neq \hat{x}_i} \sigma_i \quad (\text{C.4})$$

Here, let I_d denote the set $\{i \in I : \hat{z}_i(\omega) \neq \hat{x}_i\}$. By Assumption 5.1, $\max_{y \in Y(\hat{x}, \omega)} (q(\omega) + G(\omega)^\top \hat{x})^\top (\hat{y}(\omega) - y) < \infty$, and thus the left-hand side of the above inequality is bounded. This implies that there exists $\sigma_i \in \mathbb{R}_+, i \in I$, such that the inequality is violated for all $\bar{y} \in Y(\hat{x}, \omega)$. In other words, the following provides a sufficient condition under which it is enforced that $\hat{z}_i(\omega) = \hat{x}_i$ for all $i \in I$:

$$\sum_{i \in I_d} \sigma_i \geq \max_{y \in Y(\hat{x}, \omega)} \left\{ (q(\omega) + G(\omega)^\top \hat{x})^\top (\hat{y}(\omega) - y) \right\}. \quad (\text{C.5})$$

□

C.1.3 Proof of Theorem 5.3

Proof. By Lemma 5.2, in the reformulation (5.7), we can fix $\lambda(\omega)$ to its optimal value using the analytical form: $\lambda_i(\omega) = \sigma_i(2x_i - 1)$ for $i \in I$ and $\lambda_i(\omega) = 0$ for $i \in [n_1] \setminus I$ for every $\omega \in \Omega$. Consequently, we can substitute the bilinear terms $\lambda(\omega)^\top x$ and $\lambda(\omega)^\top z$ in the objective functions as follows: $\lambda(\omega)^\top x = \sum_{i \in I} \sigma_i(2x_i - 1)x_i = \sum_{i \in I} \sigma_i x_i$, and $\lambda(\omega)^\top z = \sum_{i \in I} \sigma_i(2x_i - 1)z_i$ for $\omega \in \Omega$. This results in the formulation (5.10). □

C.1.4 Proof of Proposition 5.4

Proof. Let $\phi(x)$ be the objective function of the reformulation (5.10). To prove the statement, it suffices to show that Algorithm 5.1 terminates with $UB = LB$ in a finite number of iterations, as this implies that $\phi(x^*) = UB = LB \leq \min_{x \in X} \phi(x) = \min_{x \in X} \{f(x) +$

$\sum_{\omega \in \Omega} p(\omega)Q(x, \omega)\}$. The last equality holds by Theorem 5.3.

For a solution $(y^l(\omega), z^l(\omega))$ and $\omega \in \Omega$, we have $\hat{Q}(x, \omega) \geq q(\omega)^\top y^l(\omega) + x^\top G(\omega)y^l(\omega) + \sum_{i \in [n_1]} \sigma_i(2x_i - 1)z_i^l(\omega)$ since the solution is feasible to the problem (5.11). Furthermore, the inequality is tight at x^l , i.e., $\hat{Q}(x^l, \omega) = q(\omega)^\top y^l(\omega) + (x^l)^\top G(\omega)y^l(\omega) + \sum_{i \in [n_1]} \sigma_i(2x_i^l - 1)z_i^l(\omega)$. Thus, the optimality cut generated using solutions $(y^l(\omega), z^l(\omega)), \omega \in \Omega$, for iteration l is a valid cut that supports the epigraph of $\sum_{\omega \in \Omega} p(\omega)\hat{Q}(x, \omega)$ at x^l , thus ensuring that the master problem exactly evaluates the objective value in the incumbent solution x^l . Since $|X| < \infty$, there exists $l < \infty$ such that $\hat{\theta} = \sum_{\omega \in \Omega} p(\omega)\hat{Q}(x^{l+1}, \omega)$ at Algorithm 5.1. This implies that $UB = LB$, as x^{l+1} is feasible to the original reformulation (5.10). \square

C.1.5 Proof of Proposition 5.5

Proof. For scenario ω and iteration l , the dual subproblem can be viewed as a restriction of the reformulated recourse problem (5.17) where variables π_k for $k = m^l + 1, \dots, m(\omega)$ are restricted to be zero. Let $(\hat{\pi}(\omega), z^l(\omega))$ be a lifted solution of $(\pi^l(\omega), z^l(\omega))$, where $\hat{\pi}_k(\omega) = \pi_k^l(\omega)$ for $k = 1, \dots, m(\omega)$ and $\hat{\pi}_k(\omega) = 0$ for $k = m^l + 1, \dots, m(\omega)$. This solution is feasible to problem (5.17), and thus the following inequality holds:

$$\begin{aligned} \hat{Q}(x, \omega) &\geq \hat{\pi}(\omega)^\top (\tilde{r}(\omega) - \tilde{T}(\omega)x) + \sum_{i \in [n_1]} \sigma_i z_i^l(\omega)(2x_i - 1) \\ &= \pi^l(\omega)^\top (r^l(\omega) - T^l(\omega)x) + \sum_{i \in [n_1]} \sigma_i z_i^l(\omega)(2x_i - 1). \end{aligned}$$

Multiplying the inequality by $p(\omega)$ and summing over all $\omega \in \Omega$ yields the optimality cut. \square

C.1.6 Proof of Proposition 5.6

Proof. Applying the Lagrangian decomposition to (5.24), we obtain the following dual

$$\min_{x \in X} \left\{ f(x) + \max_{p \in \mathcal{P}} \sum_{\omega \in \Omega} p(\omega) \left(\min_{\lambda(\omega) \in \mathbb{R}^{n_1}} -\lambda(\omega)^\top x + L(x, \lambda(\omega), \omega) \right) \right\}$$

where $L(x, \lambda(\omega), \omega) := \max\{q(\omega)^\top y + x^\top G(\omega)y : W(\omega)y + T(\omega)z = r(\omega), z \in X, y \in \mathcal{Y}\}$.

Fixing $\lambda_i(\omega) = \sigma_i(2x_i - 1)$ for $i \in [n_1]$, we obtain the formulation stated in (5.25). \square