Leveraging Decision Diagrams to Solve Two-stage Stochastic Programs with Binary Recourse and Logical Linking Constraints

Moira MacNeil*[1] and Merve Bodur†[1]

[1]Department of Mechanical and Industrial Engineering, University of Toronto

Abstract

Two-stage stochastic programs with binary recourse are challenging to solve and efficient solution methods for such problems have been limited. In this work, we generalize an existing binary decision diagram-based (BDD-based) approach of Lozano and Smith (Math. Program., 2018) to solve a special class of two-stage stochastic programs with binary recourse. In this setting, the first-stage decisions impact the second-stage constraints. Our modified problem extends the second-stage problem to a more general setting where logical expressions of the first-stage solutions enforce constraints in the second stage. We also propose a complementary problem and solution method which can be used for many of the same applications. In the complementary problem we have second-stage costs impacted by expressions of the first-stage decisions. In both settings, we convexify the second-stage problems using BDDs and parametrize either the arc costs or capacities of these BDDs with first-stage solutions depending on the problem. We further extend this work by incorporating conditional value-at-risk and we propose, to our knowledge, the first decomposition method for two-stage stochastic programs with binary recourse and a risk measure. We apply these methods to a novel stochastic dominating set problem and present numerical results to demonstrate the effectiveness of the proposed methods.

Keywords. Binary decision diagrams, two-stage stochastic programming, Benders decomposition, conditional value-at-risk

1 Introduction

Decision-making problems under uncertainty are often challenging to model and solve, especially when the decisions must be modelled as binary "yes-or-no" choices such as in the knapsack or facility location problems. One approach to address these problems is two-stage stochastic programming. This modelling framework divides decision making into two stages: first we make immediate "here-and-now" decisions without the knowledge of the value of uncertain parameters, and second, after

*m.macneil@mail.utoronto.ca
†bodur@mie.utoronto.ca

observing the outcome of the random parameters we make the remaining recourse decisions. Typically, the objective of two-stage stochastic programming is to minimize the total cost given as the first-stage cost and the expected second-stage cost. We will study a class of two-stage stochastic programs (2SP) with binary recourse and specific structure that links first-stage and second-stage decisions and their costs. Lozano and Smith first introduced binary decision diagram-based (BDD-based) methods to solve such a problem class in [29].

In this paper we generalize the family of problems studied by Lozano and Smith and extend their BDD-based methodology allowing it to be applied to a wide array of application types. We also propose an alternative to problem class that appears naturally in many of the same applications. We develop a distinct BDD-based algorithm to solve the novel problem class and show through computation that this algorithm is more efficient than that of Lozano and Smith when applied to the same problem.

In the remainder of this section we will introduce the original problem of Lozano and Smith, followed by our generalization. We will then propose our novel alternative to that problem and highlight the differences in these two classes using a variety of applications. Finally, we will summarize the contributions of this work and give the outline for the paper.

1.1 The Problem of Lozano and Smith [29]

Given $n = n_x^{\mathtt{B}} + n_x^{\mathtt{I}} + n_x^{\mathtt{C}}$ first-stage decision variables x, and real-valued objective and constraint coefficients, the two-stage stochastic programming problem considered by Lozano and Smith [29] is:

Problem 1.

$$\min\ c^\top x + \mathbb{E}[\mathcal{Q}(x,\omega)] \tag{1a}$$
$$\text{s.t. } Ax \geq b \tag{1b}$$
$$x = (x^{\mathtt{B}}, x^{\mathtt{I}}, x^{\mathtt{C}}) \in \{0,1\}^{n_x^{\mathtt{B}}} \times \mathbb{Z}^{n_x^{\mathtt{I}}} \times \mathbb{R}^{n_x^{\mathtt{C}}} \tag{1c}$$

where the value (recourse) function is

$$\mathcal{Q}(x,\omega) = \min\ d(\omega)^\top y \tag{2a}$$
$$\text{s.t. } x_i^{\mathtt{B}} = 0 \implies y \in \mathcal{W}_i(\omega) \quad \forall i = 1,\ldots,n_x^{\mathtt{B}} \tag{2b}$$
$$y \in \mathcal{Y}(\omega) \tag{2c}$$
$$y \in \{0,1\}^{n_y} \tag{2d}$$

and where ω is a random vector defined on the probability space $(\Phi, \mathcal{F}, \mathcal{P})$.

The logical constraints (2b) of the second-stage problem enforce constraints $\mathcal{W}_i(\omega)$ on the recourse variables y only if $x_i^{\mathtt{B}} = 0$. This amounts to the selection of a *single* first-stage variable $x_i^{\mathtt{B}}$ making the constraints in $\mathcal{W}_i(\omega)$ redundant.

The key insight of Lozano and Smith is that this second-stage problem can be reformulated using BDDs where the arc capacities are parameterized by the first-stage binary variable solutions, which in turn makes the problem amenable to be solved via the Benders decomposition algorithm. They use this so-called BDD-based reformulation to create a single BDD equivalent to each second-stage scenario problem, which can then be passed a first-stage candidate solution and solved efficiently via a shortest path algorithm. The shortest-path solution is then used to generate Benders cuts.

Broadly, the algorithms for solving 2SP outlined in this paper use the same mechanisms: solve the first-stage problem for a candidate solution, then reformulate the second-stage model as a BDD that incorporates some parameterization of the first-stage solutions, finally find the shortest path of the BDD using which derive a cut that will be used to generate an improved candidate solution in the first stage. However, the details of the reformulations proposed in this work are quite distinct and can lead to very different computational results.

1.2 A Generalization of Problem 1

In this work, we will consider a generalization of Problem 1 where the left-hand-side of the implication in the constraints (2b) is now a *logical expression* of the first-stage binary variables $x^{\mathtt{B}}$. We also present the first-stage problem in a more general setting where the first-stage decisions are constrained to the set $\mathcal{X}$ which does not have any restrictions. Most notably, the first-stage problem does not need to be linear, which broadens the possibilities for applications. We will use a similar procedure to that in [29] to reformulate (4) via a BDD where the *arc capacities* are parameterized using the logical functions of first-stage solutions.

Problem 2.

$$\min\ c^\top x + \mathbb{E}[\mathcal{Q}(x,\omega)] \tag{3a}$$
$$\text{s.t. } x = (x^{\mathtt{B}}, x^{\mathtt{I}}, x^{\mathtt{C}}) \in \mathcal{X} \subseteq \{0,1\}^{n_x^{\mathtt{B}}} \times \mathbb{Z}^{n_x^{\mathtt{I}}} \times \mathbb{R}^{n_x^{\mathtt{C}}} \tag{3b}$$

where

$$\mathcal{Q}(x,\omega) = \min\ d(\omega)^\top y \tag{4a}$$
$$\text{s.t. } \mathbb{I}(L_i^\omega(x)) = 1 \implies y \in \mathcal{W}_i(\omega) \quad \forall i = 1,\ldots,m_1 \tag{4b}$$
$$y \in \mathcal{Y}(\omega) \tag{4c}$$
$$y \in \{0,1\}^{n_{\mathtt{y}}}. \tag{4d}$$

In the new logical constraints (4b), the function $L_i^\omega(x)$ is a logical expression on the first-stage variables, and $\mathbb{I}(\cdot)$ is an indicator function such that $\mathbb{I}(L_i^\omega(x)) = 1$ if the expression $L_i^\omega(x)$ evaluates to true. These constraints require that for a fixed i, when $L_i^\omega(x)$ is true, the constraint set in $\mathcal{W}_i(\omega)$ is enforced for y. This logical expression is scenario-dependent as there are cases where $L_i^\omega(x)$ may be true in one scenario but not another. We remark that whereas in Problem 1 the constraint set is enforced when the first-stage binary variable $x_i^{\mathtt{B}} = 0$, in our generalization we enforce constraints when the logical expression is true, that is, the indicator returns value 1. This choice was made to remain consistent with Problem 3 which is outlined in Section 1.3.

1.3 An Alternative to Problem 2

One of the limitations of the BDD-based reformulation for the second-stage problem (4) is that the BDDs are less likely to contain isomorphic subgraphs. If the arc capacities of an otherwise isomorphic subgraph are not the same, that subgraph cannot be used to reduce the size of the BDD. Since the reformulation of (4) relies on parameterizing the arc capacities using logical functions of first-stage solutions, the resulting (reduced) BDDs are larger than if the arc capacities were all unary, as such finding the shortest path can be computationally expensive. Motivated by the search for a more computationally efficient solution approach, we present an alternative to Problem

2 that we will also reformulate using BDDs but whose reformulation does not rely on parameterized *arc capacities* and instead manipulates the *arc costs*.

Problem 3.

$$\min\ c^\top x + \mathbb{E}[\mathcal{Q}(x,\omega)] \tag{5a}$$

$$\text{s.t. } x = (x^{\mathtt{B}}, x^{\mathtt{I}}, x^{\mathtt{C}}) \in \mathcal{X} \subseteq \{0,1\}^{n_x^{\mathtt{B}}} \times \mathbb{Z}^{n_x^{\mathtt{I}}} \times \mathbb{R}^{n_x^{\mathtt{C}}} \tag{5b}$$

where

$$\mathcal{Q}(x,\omega) = \min\ (d^1(\omega) + d^2(\omega))^\top y \tag{6a}$$

$$\text{s.t. } \mathbb{I}\left(L_i^\omega(x)\right) = 1 \implies d^1_{q(i)}(\omega) = 0 \quad \forall i = 1,\ldots,m_1 \tag{6b}$$

$$y \in \mathcal{Y}(\omega) \tag{6c}$$

$$y \in \{0,1\}^{n_{\mathrm{y}}}. \tag{6d}$$

The first-stage problem is exactly that of Problem 2. The second-stage problem (6) minimizes the cost (6a) which is comprised of two parts, where we require $d^1(\omega) \geq 0$. The logical constraints (6b) enforce that if $L_i^\omega(x)$ is true, the cost of decision $y_{q(i)}$ is reduced to $d^2_{q(i)}(\omega)$ which can be 0. The BDD-based reformulation for Problem 3 is not straightforward as it requires some manipulation to be able to derive Benders cuts. This problem structure is unconventional when compared to the literature but lends itself well to many applications as we will detail next.

1.4 Applications of Problems 2 and 3

Generally, in applications for problems with the structure of Problem 2 a set of constraints can either be satisfied in the first stage or in the second stage. Whereas, applications for problems with the structure of Problem 3 appear when the first-stage decisions allow for investment in an item and in the second stage the item can be used at a reduced cost, typically for free. All the Problem 3 instances that we have encountered can also be formulated as a Problem 2 instance and vice versa but the original problem structure may naturally lend itself to one over the other. As we will see in Section 4, the BDD-based representations of (4) and (6) may also favour one problem over the other. In this section we will overview a wide variety of applications where models of the form of Problems 2 and 3 arise. Note that although we do not explicitly state it for each application, the constraint set in $\mathcal{Y}(\omega)$ typically differs between (4) and (6).

1.4.1 Stochastic shortest path

The structure of Problems 2 and 3 appears in the stochastic extensions of many classical mathematical programming problems. For example, in the stochastic shortest path (SSP) problem [37], the source is known but the destination vertex is uncertain. In the first stage, a subset of edges is selected, then the destination is observed, and in the second stage edges are selected to complete the path from source to destination. This problem is motivated by a supplier who does not know where the final destination of their products will be but would like to reserve some transportation links in advance to save on costs. Ravi and Sihna [37] prove there always exists an optimal solution to the SSP problem that is a tree. They also introduce a variation on the problem where the first-stage solution must form a tree and the edge costs are also stochastic while still forming a metric, which they call the tree-stochastic metric shortest paths (TSMSP) problem.

Both the SSP and TSMSP problems are excellent examples of our problem structure. Consider a formulation of the TSMSP problem where the variable $x_e = 1$ corresponds to selecting edge e in the first-stage tree. Then the edge costs and destination are observed and the remaining edges are selected where the variable $y_e = 1$ corresponds to selecting an edge in the second stage. Given a vertex set $\mathcal{V}$ and a subset $U \subset \mathcal{V}$, let $\delta(U)$ be the set of edges with exactly one endpoint in U. Denoting the source node s and the destination node in a given scenario t, the logical constraints (4b) in Problem 2 are for each $U \subset \mathcal{V}$ with $t \in U$ and $s \notin U$

$$\sum_{e \in \delta(U)} x_e = 0 \implies \sum_{e \in \delta(U)} y_e \geq 1.$$

That is, if an edge leaving a vertex subset is not selected in the first stage, it must be selected in the second stage, otherwise the source-destination path will not be completed. Denoting the second-stage cost of an edge as d_e, the logical constraints (6b) in the alternative Problem 3 for each edge e are

$$x_e = 1 \implies d_e = 0.$$

That is, if an edge is selected in the first stage, its second-stage cost is 0. That is, the supplier who reserved a transportation link in the first stage does not need to pay an additional cost to use it to move products in the second stage.

1.4.2 Stochastic facility location and assignment

Consider the uncapacitated facility location and assignment problem with a stochastic client set and stochastic facility opening costs [37] where in the first stage we can select some facilities in the facility set F to be opened, then observe the client set D and the costs c, then assign clients to facilities and take the recourse action of opening more facilities if required to serve all the clients.

Let variable $x_i = 1$ if facility $i \in F$ is opened in the first stage and $y_i = 1$ if it is opened in the second stage, and let $z_{ij} = 1$ if client $j \in D$ is assigned to facility $i \in F$. Then, the logical constraints in Problem 2 are for each $i \in F$ and $j \in D$

$$x_i = 0 \implies z_{ij} \leq y_i.$$

These constraints enforce if a facility is not opened in the first stage we must enforce that no client can be assigned to it unless it is opened in the second stage. In Problem 3, the logical constraints for each facility $i \in F$ are

$$x_i = 1 \implies c_i = 0,$$

that is, if a facility is opened in the first stage, it can be used in the second stage at no extra cost.

1.4.3 Stochastic knapsack

We can also apply our problem setting to stochastic knapsack problems with uncertain profit, similar to the robust version formulated in [4]. In this problem, we must first commit to procuring, i.e., producing ourselves or outsourcing, a subset of a set of items, I, with stochastic costs. These items have expected profit p_i^0 and are subject to a first-stage knapsack constraint with capacity C^0. We then observe a profit degradation d_i, and in the second stage we have three recourse options for each item. Firstly, we can choose to produce the item at the reduced profit $p_i = p_i^0 - d_i$ and consume second-stage knapsack capacity c_i. Secondly, we can repair the item consuming some extra second-stage knapsack capacity, $c_i + t_i$, but recovering the original expected profit of p_i^0, or finally, we can outsource the item so that the actual profit of the outsourced item is $p_i^0 - f_i$. The items produced or repaired in the second stage are subject to knapsack constraint with capacity C.

Let $x_i = 1$ if we select item i to procure in the first stage, and in the second stage let $y_i = 1$ if we select item i to produce ourselves and let $r_i = 1$ if we choose to repair item i. The stochastic knapsack problem can be formulated as follows

$$\max \sum_{i\in I} (p_i^0 - f_i)x_i + \mathbb{E}[\mathcal{Q}(x,\omega)] \tag{7a}$$

$$\text{s.t.} \sum_{i\in I} c_i^0 x_i \le C^0 \tag{7b}$$

$$x \in \{0,1\}^{|I|} \tag{7c}$$

where

$$\mathcal{Q}(x,\omega) = \max \sum_{i\in I} (f_i - d_i)y_i + d_i r_i \tag{8a}$$

$$\text{s.t.} \sum_{i\in I} c_i y_i + t_i r_i \le C \tag{8b}$$

$$y_i \le x_i \qquad \forall i \in I \tag{8c}$$

$$r_i \le y_i \qquad \forall i \in I \tag{8d}$$

$$y \in \{0,1\}^{|I|},\ r \in \{0,1\}^{|I|}. \tag{8e}$$

In Problem 2 the logical constraints are for each item i

$$x_i = 0 \implies y_i \le 0$$

since if item i is not selected for production in the first stage it can not be produced in the second stage. In Problem 3, the logical constraints for each item i are

$$x_i = 0 \implies f_i = 0$$

as not selecting item i for procurement in the first stage means it cannot be produced nor outsourced in the second stage and will not result in any profit.

1.4.4 Capital budgeting under uncertainty

Capital budgeting problems under uncertainty also fit this setting. We consider a stochastic investment planning problem similar to the robust optimization version of [4]. A company has allocated a budget to invest in projects with uncertain profit now or to wait and invest in them later. The early investments are incentivized by higher profits than if the company waits, where the reduced second-stage profit is denoted p_i for project i. Let $x_i = 1$ if a project i is selected in the first stage and let $y_i = 1$ if it is selected in the second stage. In Problem 2 the logical constraints for each project i are

$$x_i = 1 \implies y_i \le 0$$

that is, if a project is selected in the first stage it cannot be selected in the second stage. In Problem 3, the logical constraints project i are

$$x_i = 1 \implies p_i = 0$$

as a project selected in the first stage will not gain profits in the second stage.

1.4.5 Stochastic vertex cover

These settings appear in many well-known stochastic graph problems. For example, consider a stochastic vertex cover problem [37] on a graph $G = (\mathcal{V}, \mathcal{E}_0)$, which has many applications such as in network design. In the first stage a set of vertices is selected, then the scenario set of edges $\mathcal{E}$ and the second-stage vertex costs c_v are observed, notice that $\mathcal{E}$ may or may not be a subset of $\mathcal{E}_0$. Finally, in the second stage the vertex set is extended to a vertex cover of the edges in $\mathcal{E}$.

Let $x_v = 1$ if vertex $v \in \mathcal{V}$ is selected in the first stage, and let $y_v = 1$ if vertex $v \in \mathcal{V}$ is selected in the second stage. In Problem 2 the logical constraints are for each $\{u, v\} \in \mathcal{E} \cap \mathcal{E}_0$

$$x_v + x_u = 0 \implies y_v + y_u \geq 1,$$

that is if an edge that appears in both stages is not covered in the first stage, it must be covered in the second stage. In Problem 3, the logical constraints for each $v \in \mathcal{V}$ are

$$x_v = 1 \implies c_v = 0.$$

That is, if a vertex v is selected in the first stage, there is no additional cost to cover the edges incident to that vertex in the second stage.

1.4.6 Submodular optimization

Perhaps the most interesting potential application of these methods is within the field of submodular optimization. In their recent work Coniglio et al. study problems which maximize a concave, strictly increasing, and differentiable utility function over a discrete set of scenarios [13]. The decision maker must choose from two sets of items, the first is a set of meta items $\hat{N}$ that is linked to the set of items N via a covering relationship, i.e., an item $j \in N$ can only be selected if a meta item $\ell \in \hat{N}$ covering it is also selected. The objective function is a linear combination of submodular set functions and thus is submodular itself.

As discussed by Coniglio et al. [13], these problems arise in social network applications where influencers must be identified to rapidly spread information through a network. Well-known influence maximization models such as the independent cascade and linear threshold propagation models [24] fit this framework. Here the meta items are influencers and the items are the members of the social network and the goal is to choose a subset of influencers to maximize the expected number of members that can be reached. Coniglio et al. [13] also present an application in marketing problems where each scenario is a product to be marketed to a set of customers (items). The meta items are marketing campaigns that can only reach a subset of the customers. Here the utility function is the marginal utility of serving another unit of customer demand. Finally, these problems are also seen in stochastic facility location problems with uncertain demands, where the meta item set is the set of potential facility locations and the item set is the customers.

In our setting, we have first-stage decisions $x_\ell = 1$ if meta item $\ell \in \hat{N}$ is selected and the second-stage decisions $y_j = 1$ if item $j \in N$ is selected. Given a stochastic non-negative item rewards a_j and stochastic non-negative constant d, the second-stage objective function for a fixed scenario is

$$f\left(\sum_{j \in N} a_j y_j + d\right), \tag{9}$$

where we remark that this function is nonlinear. Here we have a pure binary optimization problem, with a nonlinear objective and linear constraints. As we are in the stochastic setting, we will take the expectation of (9) over a set of scenarios for a_j and d, each of which occur with a probability p_i. Therefore we are able to decompose the second-stage objective function into a sum of nonlinear functions, one function (9) for each scenario. This is precisely the setting of Bergman and Cire [5], who solve a binary optimization problem with linear constraints and a nonlinear objective function that can be written as the sum of auxiliary nonlinear functions. Their work models each component of the objective as a set function and then uses dynamic programming to create a decision diagram for each component. In our setting, we can model each scenario in the second stage as a decision diagram. Then in our setting the logical constraints in Problem 2 are for each item j

$$\sum_{\ell \in \hat{N}(j)} x_\ell = 0 \implies y_j = 0,$$

where we represent the set of meta items that cover item j by $\hat{N}(j)$. Thus if no meta item covering item j is selected in the first stage we cannot select item j in the second stage. In Problem 3, the logical constraints for each item j are

$$\sum_{\ell \in \hat{N}(j)} x_\ell = 0 \implies a_j = 0.$$

That is, if no meta item covering item j is selected in the first stage, no profit can be gained from j in the second stage.

1.4.7 Stochastic minimum weight dominating set

As a final example of the studied structures, in Section 7, we propose a novel stochastic dominating set problem that we call the stochastic minimum weight dominating set (SMWDS). In this problem, some vertices $v \in \mathcal{V}$ can be selected for the dominating set in the first stage, then uncertain costs and possible vertex deletions are observed and the remaining dominating set vertices are selected in the second stage.

Let $\mathcal{V}'$ denote the second-stage set of vertices, and let $\mathcal{N}'(v)$ denote the neighbourhood of vertex v in the second stage. Let $x_v = 1$ if the vertex v is selected in the first stage and let $y_v = 1$ if it is selected in the second stage. Denote the second-stage cost of a vertex v in a fixed scenario, d_v. Given this formulation, the logical constraints in Problem 2 for each $v \in \mathcal{V}'$ are

$$x_v + \sum_{u \in \mathcal{N}'(v)} x_u = 0 \implies y_v + \sum_{u \in \mathcal{N}'(v)} y_u \geq 1.$$

They enforce if neither v nor one of its second-stage neighbours is selected in the first stage, one of those vertices must be selected in the second stage. That is, if v is not dominated in the first stage, it must be dominated in the second stage. In Problem 3, the logical constraints for each $v \in \mathcal{V}'$ are

$$x_v = 1 \implies d_v = 0,$$

i.e., if vertex v is selected in the first stage, it can be used to dominate second-stage vertices at no additional cost.

1.5 Contributions and Outline

In the remainder of this paper we will present the BDD-based reformulations for the recourse structures in Problems 2 and 3 and incorporate them into a Benders decomposition algorithm. We remark that the reformulation is non-trivial for Problem 3. To demonstrate the strength of both formulations we will analyse the strength of the Benders cuts derived from the BDDs as compared to each other and to the integer L-shaped cuts. We will also extend these problems to the risk averse setting, by showing how to incorporate conditional value-at-risk (CVaR) into the models. Finally, we will examine a novel application, the SMWDS problem, and present computational results.

The major contributions of this study to the BDD and stochastic programming literature are:

- Firstly, proposing a novel BDD-based method to solving 2SP with binary recourse, this approach parametrizes BDD solutions using arc costs. We remark that these ideas have potential to be extended to other problems where a convex representation of second-stage problems is difficult to achieve, for example, 2SP with integer recourse or non-linear second-stage problems.
- Secondly, we generalize the related work of Lozano and Smith [29] to solve similar problems, which parametrizes BDD solutions using arc capacities. The generalization allows the work of these authors to be applied to problems where the second-stage constraints are not explicitly linked to a single first-stage variable.
- Thirdly, we extend our findings to a risk-averse setting, which constitutes, to our knowledge, the first decomposition method for 2SP with binary recourse and using CVaR.
- Finally, we propose the novel SMWDS problem and apply our methodologies to it.

The rest of the paper is organized as follows. In Section 2, we review the related BDD and stochastic programming literature. Then in Section 3, we give an overview of the proposed Benders decomposition algorithm. In Section 4, we reformulate the recourse problems (4) and (6) using BDDs. Next, in Section 5, we describe how to derive cuts from the BDDs that will be used in the Benders decomposition algorithms. We then compare the strength of various cuts for the decomposition algorithms. In Section 6 we extend the results to a risk-averse setting. In Section 7 we present computational results on the novel SMWDS problem. Finally, in Section 8 we conclude the paper.

2 Literature Review

Decomposition algorithms are a cornerstone of the methods for solving 2SP. Laporte and Louveaux [26] proposed the first extension of traditional Benders decomposition (i.e., the L-shaped method) to the integer setting, which they called the integer L-shaped method. They consider binary first-stage variables and complete recourse, and follow a standard branch-and-cut procedure to iteratively outer approximate second-stage value function. The integer L-shaped method is a special case of the logic-based Benders decomposition first introduced by Hooker and Ottosson [21]. We will compare our BDD-based methods to the integer L-shaped method for 2SP with binary recourse to derive logic-based Benders cuts. Carøe and Tind [11] generalize the integer L-shaped method by proposing a Benders decomposition algorithm for 2SP with continuous first-stage variables and integer recourse. They derive feasibility and optimality cuts via general duality theory which leads to a nonlinear master problem. Sherali and Fraticelli [48] use a reformulation-linearization technique or lift-and-project cutting planes to generate a partial description of the convex hull of the feasible recourse solutions for problems where the variables of both stages are binary. Sherali and Zhu [49] also use a reformulation-linearization technique or lift-and-project cutting planes in a similar way but for problems with mixed-integer variables in the first and second stage. Ahmed et al. [1] propose a finitely terminating algorithm for 2SP with mixed-integer first stage and pure-integer second-stage variables. Their algorithm exploits structural properties of the value function to avoid explicit enumeration of the search space. Zhang and Küçükyavuz [57] consider 2SP with pure-integer variables in both stages, they propose a Benders decomposition algorithm that utilizes parametric Gomory cuts to iteratively approximate the second-stage problems. Qi and Sen [36] propose an ancestral Benders cutting plane algorithm for 2SP with mixed-integer variables in both stages and where the integer variables are bounded. Li and Grossmann [27] present a Benders decomposition-based algorithm for nonlinear 2SP with mixed-binary variables in both stages. Most recently, van der Laan and Romeijnders [54] consider 2SP with mixed-integer recourse, they propose a recursive scheme to update the outer approximation of the recourse problem and to derive a new family of optimality cuts called scaled cuts.

The literature also contains many decomposition algorithms developed for the *special case* of binary first-stage variables and integer second-stage variables [3, 16, 32, 33, 34, 35, 45, 46]. A wide array of other approaches to solve stochastic integer programs exactly have been proposed including Lagranian duality-based methods [10], enumeration algorithms [41], branch-and-price schemes [30]. Most recently neural networks and supervised learning have been used to approximate the second-stage value function [15]. Interested readers may consult the introductory paper [25] and surveys [18, 40, 43, 44] on stochastic integer programming for further background.

A natural extension of stochastic programming models is to incorporate *risk measures*, especially in fields such as finance and natural disaster planning. However, the literature contains few works which incorporate risk measures into 2SP with integer second-stage variables. Schultz and Tiedemann [42] consider 2SP with CVaR and integer variables in the recourse problem. Their solution method uses Lagrangian relaxation of nonanticipativity to convexify the second-stage problems. Noyan [31] proposes the first Benders decomposition-based algorithms for 2SP with continuous second-stage problems and the CVaR risk measure, and apply these algorithms to disaster management. In their work, van Beesten and Romeijnders [53] consider mixed-integer recourse problems with CVaR, where they derive convex approximation models and corresponding error bounds to approximate the second-stage problem. Arslan and Detienne [4] study robust optimization problems where the second-stage decisions are mixed-binary. They propose a relaxation that can be solved via a branch-and-price algorithm, and give conditions under which this relaxation is exact. We will show that CVaR can be incorporated into problems with binary recourse and solved via decomposition using either BDDs or integer L-shaped cuts to convexify the second-stage problems.

We refer readers interested in background on the use of decision diagrams (DDs) in optimization to [6]. Here we will focus on the literature at the *intersection of DDs and stochastic optimization*, which has been limited to date. Haus et al. [19] consider endogenous uncertainty where decisions can impact scenario probabilities. They characterize these probabilities via a set of BDDs to obtain a mixed-integer programming (MIP) reformulation. As discussed in Section 1.1, Lozano and Smith [29] consider 2SP with pure-binary second-stage variables, they consider linear first-stage constraints and generic second-stage constraints. In the class of problems they study, selection of a first-stage binary variable makes a set of second-stage constraints redundant. They model the second-stage problems via BDDs to derive Benders cuts and demonstrate their techniques on stochastic traveling salesman problems. Serra et al. [47] propose two-stage stochastic programming for a scheduling problem, where they model both stages via multivalued DDs and link the two stages through assignment constraints to obtain an integer program. Castro et al. [12] propose a BDD-based combinatorial cut-and-lift procedure which they apply to a class of pure-binary chance-constrained problems. Most recently, Hooker [22] presents stochastic decision diagrams for *single-stage* stochastic problems by incorporating probabilities into the arcs of the DDs. These diagrams can then be used to derive optimal policies for the problem. Similar to the work of Lozano and Smith, we will consider pure-binary second-stage variables and model these problems via BDDs. We remark that these second-stage problems need not be linear (as seen in Section 1.4.6) and that the structure of the BDDs in our novel problem is distinct from those of the generalization of the work of Lozano and Smith.

3 Benders Decomposition Algorithm

We solve Problems 2 and 3 using a Benders decomposition framework and reformulate the recourse problems (4) and (6) using BDDs. In this section, we begin by giving the deterministic equivalent problem for 2SP, as it can be deemed the most generic (baseline) solution method for 2SP. Then we will give an overview of the Benders master problem and finally review the cutting plane algorithm for this decomposition.

3.1 Deterministic Equivalent Problem

As is well established in the literature as good modelling practice, in both problem types, we assume relatively complete recourse, that is, for any $x \in \mathcal{X}$ and $\omega \in \Phi$ there exists a feasible solution to (4) and to (6). We remark that in the case relatively complete recourse does not exist, feasibility cuts can easily be incorporated into the methods we propose. Following the traditional approach in two-stage stochastic programming we focus on solving the sample average approximation (SAA) problem. We assume a finite number of realizations for the vector of random variables, each given by a scenario $\omega \in \Omega$, and probability p_ω. Under such conditions, introducing scenario copies of the recourse decisions, y^ω, we are able to rewrite the two-stage problem as the monolithic deterministic equivalent problem:

$$\min\ c^\top x + \sum_{\omega\in\Omega} p_\omega (d^1_(\omega)\omega) + d^2_(\omega)\omega))^\top y^\omega \tag{10a}$$

$$\text{s.t. } x = (x^{\mathtt{B}}, x^{\mathtt{I}}, x^{\mathtt{C}}) \in \mathcal{X} \subseteq \{0,1\}^{n_x^{\mathtt{B}}} \times \mathbb{Z}^{n_x^{\mathtt{I}}} \times \mathbb{R}^{n_x^{\mathtt{C}}} \tag{10b}$$

$$\mathbb{I}\left(L_i^\omega(x)\right) = 1 \implies d^1_{q(i)}(\omega) = 0 \qquad \forall i = 1,\ldots,m_1,\ \forall \omega \in \Omega \tag{10c}$$

$$y^\omega \in \mathcal{Y}(\omega) \qquad \forall \omega \in \Omega \tag{10d}$$

$$y^\omega \in \{0,1\}^{n_y} \qquad \forall \omega \in \Omega \tag{10e}$$

Here we use the subproblem (6) to formulate the deterministic equivalent but the constraints of (4) could be substituted similarly. Typically, since the second-stage variables are binary, the problem (10) is solved as a monolithic model, or in specific cases it is solved via the integer L-shaped decomposition method. However, solving the deterministic equivalent problem does not scale well when we have a large number of scenarios and the integer L-shaped method does not have strong cuts and may be slow to converge. This motivates the search for better methods to solve stochastic programs with binary recourse. We propose decomposition methods that use BDDs to convexify the second-stage problems and derive Benders cuts that are stronger than integer L-shaped cuts.

3.2 Benders Master Problem

Given a finite set of scenarios Ω, and a set of cuts C we formulate the Benders master problem as:

$$\texttt{MP}(C) : \min\ c^\top x + \sum_{\omega \in \Omega} p_\omega \eta_\omega \tag{11a}$$

$$\text{s.t. } \mathbb{I}(L_i^\omega(x)) = 1 \implies \mu_i^\omega = 1 \qquad \forall i = 1, \dots, m_1,\ \forall \omega \in \Omega \tag{11b}$$

$$\text{(Cuts in } C) \tag{11c}$$

$$x \in \mathcal{X} \tag{11d}$$

$$\mu^\omega \in \{0,1\}^{m_1} \tag{11e}$$

The variables η_ω represent the second-stage objective function estimates. In the master problem, the constraints (11b) have been modified with the use of an indicator variable μ_i^ω. These variables are required to formulate the problem generically, as the logical expressions modify either the costs or capacities of BDD arcs, although typically, these variables and the associated constraints (11b) are not necessary. As can be seen in Section 1.4, the logical expressions of the first-stage variables are often expressions that must be equal to a binary value. In such cases we do not need the indicator variable or constraints and can directly pass these expressions to the BDDs as arc capacities or costs. simply replace it with the appropriate expression. In the remainder of this work we will use the indicator variables μ^ω to show the results both for ease of presentation and to keep the problem formulations as general as possible.

Using the indicator variables $\mu_i^\omega \in \{0,1\}$ for all $i = 1, \dots, m_1$, the constraints (6b) for scenario ω become

$$\mathbb{I}(L_i^\omega(x)) = 1 \implies \mu_i^\omega = 1 \quad \forall i = 1, \dots, m_1 \tag{12a}$$

$$\mu_i^\omega = 1 \implies d^1_{q(i)}(\omega) = 0 \quad \forall i = 1, \dots, m_1 \tag{12b}$$

and the constraints (4b) for scenario ω become

$$\mathbb{I}(L_i^\omega(x)) = 1 \implies \mu_i^\omega = 1 \quad \forall i = 1, \dots, m_1 \tag{13a}$$

$$\mu_i^\omega = 1 \implies y \in \mathcal{W}_i(\omega) \quad \forall i = 1, \dots, m_1. \tag{13b}$$

We now have that constraints (12a) and (13a) will now appear in the master problem as (11b), and constraints (12b) and (13b) remain in the respective subproblems. In the worst case, using these indicator variable will add $|\Omega|$-many variables for each of the m_1 logical constraints, however, as previously mentioned, the use of indicator variables is typically not necessary since the indicators are often simple expressions on the first-stage variables. We remark that if the technology matrix in the second stage is fixed, i.e., is not scenario dependent, we would not require the scenario index for the indicator variables.

3.3 Benders Algorithm Overview

For both of the BDD reformulation methods, the cutting plane algorithm is the same, the only difference will be in building the BDDs in the initialization step and the subproblems and therefore the cuts' structure. The details of the BDD construction are outlined in Section 4. We remark that the algorithm outlined here is a pure cutting plane version of the Benders algorithm, however it is generally implemented as a branch-and-cut algorithm as seen in Section 7.

We denote the lower bound on η_ω as LB^ω and the lower and upper bounds on the subproblem objective at iteration k as LB_k and UB_k, respectively. Let the set of cuts in the master problem be C and the cuts found at a given iteration k be C_k so that at iteration k, $C = \cup_{i=0}^{k-1} C_i$. We denote the BDD subproblems as $\texttt{SP}(\hat{\mu}, \omega)$.

0. Initialize: Set iteration counter $k = 0$ and set the upper bound $UB_0 = \infty$. Generate a BDD $\mathcal{B}^\omega$ for each scenario $\omega \in \Omega$. Let $C = C_0 = \{\eta_\omega \geq LB^\omega \ \forall \omega \in \Omega\}$.
1. Set $k = k + 1$ and solve $\texttt{MP}(C)$. If this problem is infeasible, whole problem is infeasible so terminate. Otherwise we have solution $(\hat{x}, \hat{\mu}, \hat{\eta})$, set LB_k to the objective value at this solution.
2. Solve each BDD subproblem $\texttt{SP}(\hat{\mu}, \omega)$ using the shortest path algorithm for the dual solution and yielding objective value $\hat{\tau}_\omega$. Calculate $UB_{est} = c^\top \hat{x} + \sum_{\omega \in \Omega} p_\omega \hat{\tau}_\omega$. If $UB_{est} < UB_{k-1}$, set $UB_k = UB_{est}$ and update the incumbent solution, otherwise $UB_k = UB_{k-1}$.
3. If $LB_k = UB_k$ accept the incumbent as an optimal solution and terminate. Otherwise, use the dual solutions from Step 2 to get Benders cuts C_k. Update $C = C \cup C_k$. Go to Step 1.

4 Recourse Problem Reformulation via BDD

For both Problems 2 and 3 a single BDD per scenario ω is generated in the initialization step of the algorithm. For Problem 2 these BDDs are generated for the case where $\mathbb{I}(L_i^\omega(x)) = 1$ for all $i = 1, \ldots, m_1$ and for Problem 3 they are generated under the condition where $\mathbb{I}(L_i^\omega(x)) = 0$ for all $i = 1, \ldots, m_1$. That is, they represent the set of feasible solutions to

$$\begin{aligned}
\mathcal{Q}(x, \omega) = \min\ & d(\omega)^\top y && \text{(14a)}\\
\text{s.t. } & y \in \mathcal{W}_i(\omega) \qquad \forall i = 1, \ldots, m_1 && \text{(14b)}\\
& y \in \mathcal{Y}(\omega) && \text{(14c)}\\
& y \in \{0,1\}^{n_y}, && \text{(14d)}
\end{aligned}$$

and to

$$\begin{aligned}
\mathcal{Q}(x, \omega) = \min\ & (d^1_(\omega)\omega) + d^2_(\omega)\omega))^\top y && \text{(15a)}\\
\text{s.t. } & y \in \mathcal{Y}(\omega) && \text{(15b)}\\
& y \in \{0,1\}^{n_y}, && \text{(15c)}
\end{aligned}$$

respectively. In particular for Problem 2 we are modelling the most constrained problem for each scenario, and for Problem 3 we are building BDDs with no discount for any of the variables. The structure of these BDDs will remain the same during the algorithm, having only their arc costs or capacities updated via the master problem solution $\hat{\mu}$ from one iteration to the next.

We denote a BDD for scenario ω as $\mathcal{B}^\omega = (\mathcal{N}^\omega, \mathcal{A}^\omega)$, and partition the arcs $\mathcal{A}^\omega$ into two sets: $\mathcal{A}^{\texttt{one},\omega}$ the arcs that assign a value of 1 to a second-stage variable and $\mathcal{A}^{\texttt{zero},\omega}$ the arcs that assign a value of 0 to a second-stage variable. $\mathcal{B}^\omega$ consists of m_1 arc layers $\mathcal{A}_1^\omega, \mathcal{A}_2^\omega, \dots, \mathcal{A}_{m_1}^\omega$ and of m_1+1 node layers $\mathcal{N}_1^\omega, \mathcal{N}_2^\omega, \dots, \mathcal{N}_{m_1}^\omega, \mathcal{N}_{m_1+1}^\omega$, where layer $\mathcal{N}_1^\omega$ contains only the root node $\mathtt{r}$ and layer $\mathcal{N}_{m_1+1}^\omega$ contains only the terminal node $\mathtt{t}$. We map from an arc a to a second-stage variable index j via the function $\sigma^\omega(a) = j$, for ease of presentation we assume the same variable ordering for each scenario and write the function as $\sigma(a)$. Each $\mathtt{r}$-$\mathtt{t}$ path in a BDD represents a feasible solution to the associated recourse problem by assigning values to y, and an optimal recourse solution is one corresponding to a shortest path.

Without loss of generality, we assume the variable ordering of the second-stage variables y is $1, 2, \dots, n_{\mathtt{y}}$. We denote the set of all states by $\mathcal{S}^\omega$ and the set of terminal states by $\mathcal{T}^\omega$. For a fixed scenario ω, we denote the state where the variables $1, 2, \dots, k-1$ have been assigned values but not variable k by s_k^ω. We define the state transition function $\rho_k : \mathcal{S}^\omega \setminus \mathcal{T}^\omega \times \{0,1\} \mapsto \mathcal{S}^\omega$, where $s_{k+1}^\omega = \rho_k(s_k^\omega, \hat{y}_k)$, where $\hat{y}_k$ is the value assigned to variable y_k. Finally, we denote the set of remaining second-stage solutions that can be explored given a state s_k^ω and variable assignment $\hat{y}_k$, $\bar{Y}^\omega(s_k^\omega, \hat{y}_k)$. We will denote the BDDs resulting from Problems 2 and 3 as `BDD2` and `BDD3`, respectively.

4.1 `BDD2` Reformulation

Here we restate the method of Lozano and Smith [29], which allows us to reformulate the generalized recourse problem (4) by building a BDD where some arcs have capacities parameterized by the master problem solution. We remark that when the variables (or expressions of variables) of the first-stage problem can be used directly as the indicators these variables (or expressions) will appear as the arc capacities.

4.1.1 `BDD2` Representation

The procedure to build `BDD2` is given in Algorithm 1. It begins by building the BDD in the natural way, under the assumption that all second-stage constraints are enforced, using the transition function to move to a new state for each value choice for a variable y_j. When a state, s^{j+1}, becomes infeasible after assigning variable y_j value $\hat{y}_j$, we first check if the cause of the infeasibility is due to the constraint set in $\mathcal{Y}(\omega)$. That is if $\bar{Y}^\omega(s^j, \hat{y}_j) \cap \mathcal{Y}(\omega) = \emptyset$, then no matter the value of the first-stage solution, every solution in $\bar{Y}^\omega(s^j, \hat{y}_j)$ is infeasible. However, if the state is infeasible but $\bar{Y}^\omega(s^j, \hat{y}_j) \cap \mathcal{Y}(\omega) \neq \emptyset$, then there exists some set of violated constraints with indices in $\mathcal{I}^\omega = \{i : \bar{Y}^\omega(s^j, \hat{y}_j) \cap \mathcal{W}_i(\omega) = \emptyset\}$ corresponding to indicator variables μ_i^ω. The new state is defined by the transition function and by setting $\mu_i^\omega = 0$ for all $i \in \mathcal{I}^\omega$, that is, we update the state defined by the transition function to reflect the second-stage constraints $y \in \mathcal{W}_i(\omega)$ being made redundant. We then add capacities μ_i^ω for all $i \in \mathcal{I}^\omega$ to the arc corresponding to the value $\hat{y}_j$. We remark that a single arc may have multiple capacities, one for each i in $\mathcal{I}^\omega$, all of which must equal one in order to utilize this arc. This ensures that the new state is only achievable when the violated constraints have been made redundant by the first-stage solutions. After the BDD is created, it is reduced so that no pair of nodes in a layer are equivalent, see [6, 55] for more information about BDD reductions.

Example 1. We will use the SMWDS as a running example throughout this paper. An IP formu-

Algorithm 1: `BDD2` construction

Input: Transition functions ρ_j, scenario ω
Output: A reduced, exact BDD for (4)

Create root node $\mathtt{r}$ with initial state $s^{\omega}_{\mathtt{r}}$;
forall $j \in \{1, 2, \ldots, m_1\}$, $u \in \mathcal{N}^{\omega}_j$ **do**
  for $b \in \{0, 1\}$ **do**
    Update $\bar{Y}^{\omega}(s^{\omega}_u, b)$;
    if $\rho_j(s^{\omega}_u, b)$ *is feasible* **then**
      Create node $v \in \mathcal{N}^{\omega}_{j+1}$ with state $\rho_j(s^{\omega}_u, b)$;
      if $b = 1$ **then**
        Add uncapacitated arc $a = (u, v) \in \mathcal{A}^{\mathtt{one},\omega}$ with cost $d_{\sigma(a)}(\omega)$ to $\mathcal{A}^{\omega}_j$;
      else
        Add uncapacitated arc $(u, v) \in \mathcal{A}^{\mathtt{zero},\omega}$ with cost 0 to $\mathcal{A}^{\omega}_j$;
    else
      if $\bar{Y}^{\omega}(s^{\omega}_u, b) \cap \mathcal{Y}(\omega) \neq \emptyset$ **then**
        Find set of violated constraint indices $\mathcal{I}^{\omega} = \{i : \bar{Y}^{\omega}(s^{\omega}_u, b) \cap \mathcal{W}_i(\omega) = \emptyset\}$;
        Create node $v \in \mathcal{N}^{\omega}_{j+1}$ with state $\rho_j(s^{\omega}_u, b)$ and ignoring constraints $\mathcal{W}_i(\omega)\ \forall i \in \mathcal{I}^{\omega}$;
        if $b = 1$ **then**
          Add arc $a = (u, v) \in \mathcal{A}^{\mathtt{one},\omega}$ with cost $d_{\sigma(a)}(\omega)$ and capacities $(1 - \mu^{\omega}_i)\ \forall i \in \mathcal{I}^{\omega}$ to $\mathcal{A}^{\omega}_j$;
        else
          Add arc $(u, v) \in \mathcal{A}^{\mathtt{zero},\omega}$ with cost 0 and capacities $(1 - \mu^{\omega}_i)\ \forall i \in \mathcal{I}^{\omega}$ to $\mathcal{A}^{\omega}_j$;
Reduce the resulting BDD

lation for the recourse problem of SMWDS is:

$$\mathcal{Q}(x, \omega) = \min \sum_{v \in \mathcal{V}'} d_v(\omega) y_v \tag{16a}$$

$$\text{s.t. } y_v + \sum_{u \in \mathcal{N}'(v)} y_u \geq 1 - \left(\hat{x}_v + \sum_{u \in \mathcal{N}'(v)} \hat{x}_u \right) \quad \forall\, v \in \mathcal{V}' \tag{16b}$$

$$y_v \in \{0, 1\} \quad \forall\, v \in \mathcal{V}' \tag{16c}$$

where the sets $\mathcal{N}'(v)$ and $\mathcal{V}'$ are as defined in Section 1.4.7. For illustrative purposes, we consider an instance with a single scenario whose scenario graph is pictured in Figure 1. The indicator constraints for this problem are equivalent to the IP constraints (16b), enforce that if a vertex $v \in \mathcal{V}'$ is not dominated in the first stage it must be dominated in the second stage. A more formal description of the entire SMWDS problem is given in Section 7.

We also remark that we do not need to introduce an indicator variable for this problem and instead use a summation of the appropriate first-stage variables. For a vertex v, if $x_v + \sum_{u \in \mathcal{N}'(v)} x_u = 0$ then we must enforce covering constraints in the second stage. We use this expression as the negation of the indicator $(1 - \mu^{\omega}_v)$. The `BDD2` for this scenario is pictured in Figure 2a, the state at each node represents the vertices that still need to be dominated, and each solid arc has a weight of one. We can see that if we select $y_0 = y_1 = 0$, under the assumption that all second-stage constraints are enforced, if we also set $y_2 = 0$ we transition to an infeasible state.

This state is infeasible because we have $\hat{x}_v = 0$ for all $v \in \mathcal{V}'$ while constructing the BDD, and $y_0 = y_1 = y_2 = 0$ and the domination constraint (16b) for vertex 1 is violated since

$$y_0 + y_1 + y_2 < 1 - \hat{x}_0 - \hat{x}_1 - \hat{x}_2.$$

However, the violated constraint in this infeasible state is the logical constraint and not the other constraints on the second-stage variables, i.e., we have violated a constraint in the set $\mathcal{W}_1(\omega)$ not in the set $\mathcal{Y}(\omega)$. Thus, we could have selected one of x_0, x_1, or x_2 to dominate vertex 1 in the first stage and this state would no longer be infeasible. To model this in the BDD, we add a node in the next layer with state $0, 2, 3, 4$, i.e., assuming vertex 1 is dominated in the first stage, and add an arc for the decision $y_2 = 0$ to this new node with capacity $x_0 + x_1 + x_2$. The arc capacity now enforces that if $\hat{x}_0 = \hat{x}_1 = \hat{x}_2 = 0$, there can be no flow on the arc and the new state is unreachable.

In this example, we can also see that there may be multiple capacity constraints on a single arc, such as in the last arc layer of the BDD. Finally, we remark that due to the arcs having different capacities, we are unable to reduce the BDD any further.

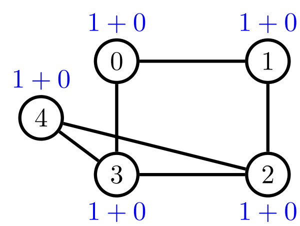


Figure 1: A single scenario for the SMWDS problem, where the indices are given inside the vertices, and the vertex weights ($d(\omega) = d^1(\omega) + d^2(\omega)$) are outside the vertices.

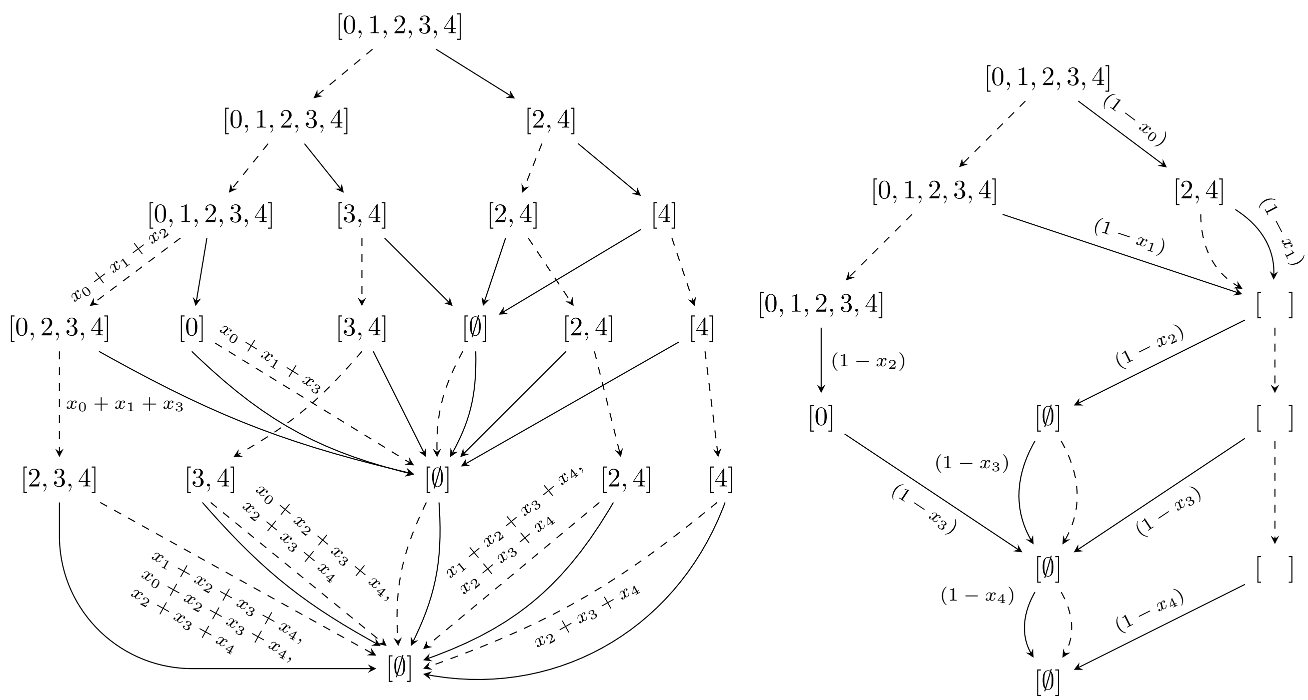
4.1.2 `BDD2` Recourse Problem Reformulation

Given a scenario ω and a first-stage solution $\hat{\mu}^\omega$, the `BDD2` reformulation of (4) yields a minimum cost network flow problem $\texttt{P2}(\hat{\mu}^\omega, \omega)$ with capacity constraints (17e) on some arcs of the BDD. In the linear programming formulation (17) we have continuous decision variables f_a that give the flow over arc $a \in \mathcal{A}^\omega$ and we define the set of indices $\mathcal{I}_a^\omega$ for an arc $a \in \mathcal{A}^\omega$ such that there can be no flow on a unless $\hat{\mu}_i^\omega = 0$ for all $i \in \mathcal{I}_a^\omega$.

$$\texttt{P2}(\hat{\mu}^\omega, \omega) : \min \sum_{a \in \mathcal{A}^{\texttt{one},\omega}} f_a d_{v(a)}(\omega) \tag{17a}$$

$$\text{s.t.} \sum_{a=(n,k) \in \mathcal{A}^\omega} f_a - \sum_{a=(k,n) \in \mathcal{A}^\omega} f_a = 0 \quad \forall n \in \mathcal{N}^\omega \setminus \{\texttt{r}, \texttt{t}\} \tag{17b}$$

$$\sum_{a=(\texttt{r},j) \in \mathcal{A}^\omega} f_a = 1 \tag{17c}$$



(a) `BDD2`, where arc capacities are given.

(b) `BDD3`, where arc costs are given.

Figure 2: `BDD2` and `BDD3` reformulations for the instance in Figure 1.

$$\sum_{a=(j,\mathtt{t})\in\mathcal{A}^\omega} f_a = 1 \tag{17d}$$

$$f_a \leq 1 - \hat{\mu}_i^\omega \qquad \forall\ a \in \mathcal{A}^\omega, i \in \mathcal{I}_a^\omega \tag{17e}$$

$$f_a \geq 0 \qquad \forall\ a \in \mathcal{A}^\omega \tag{17f}$$

To derive traditional Benders cuts from this problem we take the dual of the modified minimum cost network flow problem, introducing variables π and β for the flow conservation and capacity constraints respectively, resulting in the following model

$$\mathtt{D2}(\hat{\mu}^\omega,\omega) \max\ \pi_{\mathtt{r}} - \pi_{\mathtt{t}} - \sum_{a\in\mathcal{A}^\omega}\sum_{i\in\mathcal{I}_a^\omega} \beta_{ai}(1-\hat{\mu}_i^\omega) \tag{18a}$$

$$\text{s.t. } \pi_u - \pi_v - \sum_{i\in\mathcal{I}_a^\omega} \beta_{ai} \leq 0 \qquad \forall a=(u,v)\in\mathcal{A}^{\mathtt{zero},\omega} \tag{18b}$$

$$\pi_u - \pi_v - \sum_{i\in\mathcal{I}_a^\omega} \beta_{ai} \leq d_{\sigma(a)}(\omega) \quad \forall a=(u,v)\in\mathcal{A}^{\mathtt{one},\omega} \tag{18c}$$

$$\beta_{ai} \geq 0 \qquad \forall\ a\in\mathcal{A}^\omega, \forall i \in \mathcal{I}_a^\omega. \tag{18d}$$

Without loss of generality, we can assume $\pi_{\mathtt{t}} = 0$ as the constraint of the primal associated with this variable is linearly dependant on the other constraints.

Proposition 1. *Given a first-stage solution $\hat{\mu}^\omega$ and a scenario ω,* P2$(\hat{\mu}^\omega, \omega)$ *is equivalent to* (4).

Proof. Each `r-t` path in the BDD corresponds to exactly one feasible solution of (4) with equal objective function, and vice versa. $\square$

4.2 `BDD3` Reformulation

The second BDD reformulation builds BDDs where the first-stage solutions impact the arc weights and allows for a convex representation of subproblem (6).

4.2.1 `BDD3` Representation

The procedure to create a BDD of type `BDD3` is given in Algorithm 2. The only difference from the creation of a standard BDD for the given transition function is that when we create an arc corresponding to a decision assigning $y_{\sigma(a)} = 1$, we give the arc cost $d^1_{\sigma(a)}(\omega)(1 - \hat{\mu}^\omega_{\sigma(a)}) + d^2_{\sigma(a)}(\omega)$. This ensures that when a first-stage decision is $\hat{\mu}^\omega_{\sigma(a)} = 1$, the cost to make decision $y_{\sigma(a)}$ is $d^2_{\sigma(a)}(\omega)$ otherwise the cost to make decision $y_{\sigma(a)}$ is $d^1_{\sigma(a)}(\omega) + d^2_{\sigma(a)}(\omega)$. We remark that all the arcs in this BDD are uncapacitated.

Algorithm 2: `BDD3` construction

Input: Transition functions ρ_j, scenario ω
Output: A reduced, exact BDD (6)
Create root node `r` with initial state;
forall $j \in \{1, 2, \ldots, m_1\}$, $u \in \mathcal{N}^\omega_j$ **do**
  for $b \in \{0, 1\}$ **do**
    if $\rho_j(s^\omega_u, b)$ *is not infeasible* **then**
      Create node $v \in \mathcal{N}^\omega_{j+1}$ with state $\rho_j(s^\omega_u, b)$;
      if $b = 1$ **then**
        Add arc $a = (u, v) \in \mathcal{A}^{\texttt{one},\omega}$ with cost $d^1_{\sigma(a)}(\omega)(1 - \hat{\mu}^\omega_{\sigma(a)}) + d^2_{\sigma(a)}(\omega)$ to $\mathcal{A}^\omega_j$;
      else
        Add arc $(u, v) \in \mathcal{A}^{\texttt{zero},\omega}$ with cost 0 to $\mathcal{A}^\omega_j$;
Reduce the resulting BDD

Example 2. Recall the scenario in Figure 1, to construct `BDD3` we proceed as if we were building a dominating set BDD. Again, the state at each node represents the vertices that still need to be dominated, each dashed arc has a weight of zero, and because each vertex has cost $d^1_i(\omega) + d^2_i(\omega) = 1 + 0$ each solid arc will have a weight of $(1 - \hat{x}_i)$, that is, if $\hat{x}_i = 1$, then $d^1_i(\omega) = 0$. The reduced `BDD3` can be seen in Figure 2b, where nodes that were merged during the reduction have no state shown. We remark that this BDD has fewer nodes and has a smaller width than the BDD created using the `BDD2` procedure for the same instance.

4.2.2 `BDD3` Recourse Problem Reformulation

Given a fixed scenario ω and a master problem solution $\hat{\mu}^\omega$, the BDD reformulation of (6) yields a minimum cost network flow problem, P3$(\hat{\mu}^\omega, \omega)$, with modified objective functions over the BDD.

Again, let continuous decision variables f_a be the flow over arc $a \in \mathcal{A}^\omega$.

$$\begin{aligned}
\texttt{P3}(\hat{\mu}^\omega, \omega) : \min \quad & \sum_{a \in \mathcal{A}^{\texttt{one},\omega}} f_a \left(d^1_{\sigma(a)}(\omega)(1 - \hat{\mu}^\omega_{\sigma(a)}) + d^2_{\sigma(a)}(\omega) \right) && && \text{(19a)}\\
\text{s.t.} \quad & \sum_{a=(n,k) \in \mathcal{A}^\omega} f_a - \sum_{a=(k,n) \in \mathcal{A}^\omega} f_a = 0 && \forall n \in \mathcal{N}^\omega \setminus \{\texttt{r}, \texttt{t}\} && \text{(19b)}\\
& \sum_{a=(\texttt{r},j) \in \mathcal{A}^\omega} f_a = 1 && && \text{(19c)}\\
& \sum_{a=(j,\texttt{t}) \in \mathcal{A}^\omega} f_a = 1 && && \text{(19d)}\\
& f_a \geq 0 && \forall\ a \in \mathcal{A}^\omega && \text{(19e)}
\end{aligned}$$

Deriving cuts from this reformulated subproblem is not as straightforward as in the case of BDD2. In the next section, we will describe in detail how to derive Benders cuts from $\texttt{P3}(\hat{\mu}^\omega, \omega)$.

4.2.3 An Alternative Recourse Problem Reformulation for BDD3

Traditionally, to derive Benders cuts using the BDD reformulation, we take the dual of (19) which results in the following:

$$\begin{aligned}
\texttt{D3}(\hat{\mu}^\omega, \omega) \max \ & \pi_{\texttt{r}} - \pi_{\texttt{t}} && && \text{(20a)}\\
\text{s.t. } & \pi_i - \pi_j \leq 0 && \forall (i,j) \in \mathcal{A}^{\texttt{zero},\omega} && \text{(20b)}\\
& \pi_i - \pi_j \leq \left(d^1_{\sigma(a)}(\omega)(1 - \hat{\mu}^\omega_{\sigma(a)}) + d^2_{\sigma(a)}(\omega) \right) && \forall a = (i,j) \in \mathcal{A}^{\texttt{one},\omega} && \text{(20c)}
\end{aligned}$$

However, Benders cuts cannot immediately be taken from this dual problem since the first-stage variables appear in the constraints, not the objective. We will reformulate the primal problem into $\overline{\texttt{P3}}(\hat{\mu}^\omega, \omega)$ to ensure the first-stage variables appear only in the objective of the dual subproblem and show at an optimal solution these problems are equivalent in the sense that they achieve the same objective value for the same master problem solution.

Introducing a new set of arc variables, γ, we propose the following extended minimum cost network flow formulation:

$$\begin{aligned}
\overline{\texttt{P3}}(\hat{\mu}^\omega, \omega) : \min \quad & \sum_{a \in \mathcal{A}^{\texttt{one},\omega}} \left(d^1_{\sigma(a)}(\omega) + d^2_{\sigma(a)}(\omega) \right) f_a + \sum_{a \in \mathcal{A}^{\texttt{one},\omega}} d^2_{\sigma(a)}(\omega) \gamma_a && && \text{(21a)}\\
\text{s.t.} \quad & \sum_{a=(\texttt{r},j) \in \mathcal{A}^\omega} f_a + \sum_{a=(\texttt{r},j) \in \mathcal{A}^{\texttt{one},\omega}} \gamma_a = 1 && && \text{(21b)}\\
& \sum_{a=(i,\texttt{t}) \in \mathcal{A}^\omega} f_a + \sum_{a=(i,\texttt{t}) \in \mathcal{A}^{\texttt{one},\omega}} \gamma_a = 1 && && \text{(21c)}\\
& \sum_{\substack{a=(i,j) \\ \in \mathcal{A}^\omega}} f_a + \sum_{\substack{a=(i,j) \\ \in \mathcal{A}^{\texttt{one},\omega}}} \gamma_a = \sum_{\substack{a=(j,i) \\ \in \mathcal{A}^\omega}} f_a + \sum_{\substack{a=(j,i) \\ \in \mathcal{A}^{\texttt{one},\omega}}} \gamma_a && \forall i \in \mathcal{N} && \text{(21d)}\\
& \gamma_a \leq \hat{\mu}^\omega_{\sigma(a)} && \forall a \in \mathcal{A}^{\texttt{one},\omega} && \text{(21e)}\\
& \gamma_a \geq 0 && \forall a \in \mathcal{A}^{\texttt{one},\omega} && \text{(21f)}\\
& f_a \geq 0 && \forall a \in \mathcal{A}^\omega && \text{(21g)}
\end{aligned}$$

Proposition 2. *Given a scenario ω and binary first-stage decisions $\hat{\mu}^\omega$, formulations $\mathtt{P3}(\hat{\mu}^\omega,\omega)$ and $\overline{\mathtt{P3}}(\hat{\mu}^\omega,\omega)$ have the same optimal objective value.*

Proof. Denote an optimal solution to $\mathtt{P3}(\hat{\mu}^\omega,\omega)$ as $\hat{f}^1$ and let τ_1^* denote the optimal objective value. Similarly, denote an optimal solution to $\overline{\mathtt{P3}}(\hat{\mu}^\omega,\omega)$ as $(\hat{f}^2,\hat{\gamma}^2)$ and let τ_2^* denote the optimal objective value.

1. Given $\hat{f}^1$, let $\gamma_a^2 = \hat{f}_a^1 \hat{\mu}_{\sigma(a)}^\omega$ and $f_a^2 = \hat{f}_a^1 - \gamma_a^2$ for all $a \in \mathcal{A}^{\mathtt{one},\omega}$ and let $f_a^2 = \hat{f}_a^1$ for all $a \in \mathcal{A}^{\mathtt{zero},\omega}$. We will first show that the constructed solution (f^2,γ^2) is feasible to (21). For each arc $a \in \mathcal{A}^{\mathtt{zero},\omega}$, we have flow $f_a^2 = \hat{f}_a^1$. For each arc $a \in \mathcal{A}^{\mathtt{one},\omega}$, we have if $\hat{\mu}_{\sigma(a)}^\omega = 0$ then $\gamma_a^2 = 0$ and $f_a^2 = \hat{f}_a^1$, otherwise, if $\hat{\mu}_{\sigma(a)}^\omega = 1$ then $\gamma_a^2 = \hat{f}_a^1$ and $f_a^2 = 0$. In either case the flow on each $a \in \mathcal{A}^{\mathtt{one},\omega}$ is $f_a^2 + \gamma_a^2 = \hat{f}_a^1$. Therefore any solution feasible to (19) is also feasible to (21b)-(21d). Constraints (21e) hold since by construction if $\hat{\mu}_{\sigma(a)}^\omega = 0$ then $\gamma_a^2 = 0$, and there is an implicit upper bound of 1 on f_a^1 at any optimal solution, therefore if $\hat{\mu}_{\sigma(a)}^\omega = 1$, $\gamma_a^2 \le 1$ by construction. Similarly, by constraint (19e), $\hat{f}^1 \ge 0$ and $\hat{\mu}_{\sigma(a)}^\omega \ge 0$, so $\gamma_a^2 \ge 0$ and $f_a^2 \ge 0$ by construction, since $\hat{\mu}_{\sigma(a)}^\omega \le 1$, and constraints (21f) and (21g) hold. Thus given the solution $\hat{f}^1$ we can construct a feasible solution to $\overline{\mathtt{P3}}(\hat{\mu}^\omega,\omega)$.

   We will now show that at the constructed solution (f^2,γ^2), the objective values of (19) and (21) are equal. The objectives of both (19) and (21) consider only arcs in $\mathcal{A}^{\mathtt{one},\omega}$, so we compare terms for a fixed arc $a \in \mathcal{A}^{\mathtt{one},\omega}$ at binary solutions of $\hat{\mu}_{\sigma(a)}^\omega$. As before, if $\hat{\mu}_{\sigma(a)}^\omega = 0$, $\gamma_a^2 = 0$ by construction and $f_a^2 = \hat{f}_a^1$. Thus the objective term in $\overline{\mathtt{P3}}(\hat{\mu}^\omega,\omega)$ at arc a is $(d_{\sigma(a)}^1(\omega) + d_{\sigma(a)}^2(\omega)\hat{f}_a^1$, the same as in $\mathtt{P3}(\hat{\mu}^\omega,\omega)$. Again, by construction, if $\hat{\mu}_{\sigma(a)}^\omega = 1$ then $\gamma_a^2 = \hat{f}_a^1$ and $f_a^2 = 0$. Thus, the objective term for arc a in $\overline{\mathtt{P3}}(\hat{\mu}^\omega,\omega)$ is $d_{\sigma(a)}^2(\omega)\hat{f}_a^1$, which is equal to the objective term of $\mathtt{P3}(\hat{\mu}^\omega,\omega)$ since we have $f_a^1(1-\hat{\mu}_{\sigma(a)}^\omega)d_{\sigma(a)}^1(\omega) = 0$. The constructed solution to $\overline{\mathtt{P3}}(\hat{\mu}^\omega,\omega)$ yields the objective value of τ_1^* and we conclude $\tau_1^* \ge \tau_2^*$.

2. Given $(\hat{f}^2,\hat{\gamma}^2)$, let $f_a^1 = \hat{f}_a^2 + \hat{\gamma}_a^2$ for all $a \in \mathcal{A}^{\mathtt{one},\omega}$ and let $f_a^1 = \hat{f}_a^2$ for all $a \in \mathcal{A}^{\mathtt{zero},\omega}$.

   We will show the solution, f^1, is feasible to (19). For each arc $a \in \mathcal{A}^{\mathtt{one},\omega}$, if $\hat{\mu}_{\sigma(a)}^\omega = 0$, by constraint (21e) $\hat{\gamma}_a^2 = 0$, thus $f_a^1 = \hat{f}_a^2$. If $\hat{\mu}_{\sigma(a)}^\omega = 1$ then $f_a^1 = \hat{f}_a^2 + \hat{\gamma}_a^2$ but at an optimal solution of $\overline{\mathtt{P3}}(\hat{\mu}^\omega,\omega)$ since $d_{\sigma(a)}^1(\omega) > 0$ by definition and we are minimizing we will have $\hat{\gamma}_a^2 = 1$ and $\hat{f}_a^2 = 0$, thus $f_a^1 = \hat{\gamma}_a^2$. For each arc $a \in \mathcal{A}^{\mathtt{zero},\omega}$, we have $f_a^1 = \hat{f}_a^2$. This solution is feasible to constraints (19b)-(19d) since the flow on an arc is either given by $\hat{\gamma}_a^2$ or $\hat{f}_a^2$ and thus satisfy flow balance. Constraints (19e) hold since by (21f) and (21g) $\hat{f}_a^2 \ge 0$ and $\hat{\gamma}_a^2 \ge 0$ so we have $f_a^1 \ge 0$ by construction.

   We will again compare objective terms for a fixed arc $a \in \mathcal{A}^{\mathtt{one},\omega}$ at binary solutions of $\hat{\mu}_{\sigma(a)}^\omega$. As before, if $\hat{\mu}_{\sigma(a)}^\omega = 0$, $f_a^1 = \hat{f}_a^2$ and the objective term for $\mathtt{P3}(\hat{\mu}^\omega,\omega)$ at a is $(d_{\sigma(a)}^1(\omega) + d_{\sigma(a)}^2(\omega))\hat{f}_a^2$, the same as in $\overline{\mathtt{P3}}(\hat{\mu}^\omega,\omega)$. If $\hat{\mu}_{\sigma(a)}^\omega = 1$, again, we have $f_a^1 = \hat{\gamma}_a^2$. The objective term of $\overline{\mathtt{P3}}(\hat{\mu}^\omega,\omega)$ is $d_{\sigma(a)}^2(\omega)\hat{\gamma}_a^2$, which is equal to the objective term of $\mathtt{P3}(\hat{\mu}^\omega,\omega)$ since we have $f_a^1(d_{\sigma(a)}^1(\omega)(1 - \hat{\mu}_{\sigma(a)}^\omega) + d_{\sigma(a)}^2(\omega)) = \hat{\gamma}_a^2 d_{\sigma(a)}^2(\omega)$.

   The constructed solution to $\mathtt{P3}(\hat{\mu}^\omega,\omega)$ yields the objective value of τ_2^* and we conclude $\tau_1^* \le \tau_2^*$.

Therefore we have shown at a binary first-stage decision $\hat{\mu}^\omega$, the formulations $\texttt{P3}(\hat{\mu}^\omega,\omega)$ and $\overline{\texttt{P3}}(\hat{\mu}^\omega,\omega)$ are have the same optimal objective value, as required. $\square$

We remark that Proposition 2 relies on binary $\hat{\mu}^\omega$ and does not necessarily hold for fractional values of $\hat{\mu}^\omega$. We also remark that the proof of Proposition 2, not only shows the equality of the objective functions of $\texttt{P3}(\hat{\mu}^\omega,\omega)$ and $\overline{\texttt{P3}}(\hat{\mu}^\omega,\omega)$ at optimality but gives a transformation between the solutions of these two formulations. That is, given an optimal solution to one formulation, we can construct an optimal solution to the other.

By Proposition 2, we can derive Benders cuts from the dual of (21):

$$\overline{\texttt{D3}}(\hat{\mu}^\omega,\omega): \max\ \pi_{\texttt{r}} - \pi_{\texttt{t}} - \sum_{a\in\mathcal{A}^{\texttt{one},\omega}} z_a\hat{\mu}^\omega_{\sigma(a)} \tag{22a}$$

$$\text{s.t. } \pi_i - \pi_j \le 0 \qquad \forall (i,j)\in\mathcal{A}^{\texttt{zero},\omega} \tag{22b}$$

$$\pi_i - \pi_j \le d^1_{\sigma(a)}(\omega) + d^2_{\sigma(a)}(\omega) \quad \forall a=(i,j)\in\mathcal{A}^{\texttt{one},\omega} \tag{22c}$$

$$\pi_i - \pi_j \le z_a + d^2_{\sigma(a)}(\omega) \qquad \forall a=(i,j)\in\mathcal{A}^{\texttt{one},\omega} \tag{22d}$$

$$z_a \ge 0 \qquad \forall a=(i,j)\in\mathcal{A}^{\texttt{one},\omega} \tag{22e}$$

Again, without loss of generality, we can assume $\pi_{\texttt{t}} = 0$ as the constraint of the primal associated with this variable is linearly dependent on the other constraints.

Proposition 3. *Given a scenario ω and a binary first-stage solution $\hat{\mu}^\omega$, $\overline{\texttt{P3}}(\hat{\mu}^\omega,\omega)$ and (6) have the same optimal objective value.*

Proof. First observe that every `r-t` path in the BDD resulting from the `BDD3` procedure corresponds to exactly one feasible solution with the same objective value as in (6). Every feasible solution in (6) corresponds to exactly one `r-t` path in the BDD with the same length as the objective value. That is, $\texttt{P3}(\hat{\mu}^\omega,\omega)$ is equivalent to (6).

We also have by the proof of Proposition 2, given an optimal solution to $\overline{\texttt{P3}}(\hat{\mu}^\omega,\omega)$ we can construct an optimal solution to $\texttt{P3}(\hat{\mu}^\omega,\omega)$ and vice versa. Therefore, by transitivity, we have $\overline{\texttt{P3}}(\hat{\mu}^\omega,\omega)$ and (6) have the same optimal objective value as required. $\square$

5 Benders Cuts

In this section, we will explain how to derive cuts and then strengthen the cuts derived from both BDD-based methods. We then provide details of the integer L-shaped method, which is a well-known method in the stochastic programming literature and will serve as a point of comparison to the BDD-based decompositions. Next, we describe cuts based on the linear programming (LP) relaxation of the subproblem that can be added to any formulation. Finally, we compare the strength of all the presented cuts.

5.1 BDD-based Benders Cuts

For a fixed scenario ω the Benders cuts that come from `BDD2` are:

$$\eta_\omega \ge \hat{\pi}_{\texttt{r}} - \sum_{a\in\mathcal{A}^\omega}\sum_{i\in\mathcal{I}^\omega_a}\hat{\beta}_{ai}(1-\mu^\omega_i). \tag{23}$$

For a scenario BDD $\mathcal{B}^\omega$ and master problem solution $\hat{\mu}^\omega$ we calculate the dual solutions using a bottom-up shortest-path algorithm, as described in [29]. The terminal has dual value $\pi_{\mathtt{t}} = 0$ and each subsequent BDD node u has dual value π_u, the length of the shortest path from u to $\mathtt{t}$. The β_{ai} variables are only on those arcs with capacities, for such an arc $a = (u, v)$ if $\pi_u - \pi_v$ is shorter than the arc length, that is, the shortest path does not lie on a, then $\beta_{ai} = 0$ for all $i \in \mathcal{I}_a^\omega$. Otherwise, if $a \in \mathcal{A}^{\mathtt{zero},\omega}$ we have $\pi_u - \pi_v > 0$, so for some arbitrary $\bar{i} \in \mathcal{I}_a^\omega$ where $\hat{\mu}_{\bar{i}}^\omega = 1$ we set $\beta_{a\bar{i}} = \pi_u - \pi_v$ and $\beta_{ai} = 0$ for all $i \neq \bar{i} \in \mathcal{I}_a^\omega$. Similarly, if $a \in \mathcal{A}^{\mathtt{one},\omega}$ we have $\pi_u - \pi_v - d_{\sigma(a)}(\omega) > 0$, then we set $\beta_{a\bar{i}} = \pi_u - \pi_v - d_{\sigma(a)}(\omega)$ for an arbitrary $\bar{i} \in \mathcal{I}_a^\omega$ where $\hat{\mu}_{\bar{i}}^\omega = 1$, and set $\beta_{ai} = 0$ for all $i \neq \bar{i} \in \mathcal{I}_a^\omega$. These dual values can be computed in linear time with respect to the number of arcs in the BDD.

Example 3. Recall, in Example 1, we built the BDD for the scenario graph in Figure 1, we will now use this BDD to compute a cut for a master problem solution $\hat{x} = [0, 0, 1, 0, 0]$. This solution has $x_2 = 1$, thus any arc where x_2 appears in all the capacity expressions will not be interdicted. In Figure 3a those arcs which still have capacity 0 are highlighted. We are then able to compute the dual values π using the bottom up shortest path algorithm and obtain $\pi_{\mathtt{r}} = 1$. The dual values of each node are given inside the node in Figure 3a. Finally, to derive the cut we must compute the values of β_{ai}. We observe that to have $\beta_{ai} \neq 0$ we must have $\hat{\mu}_i^\omega = 1$, in our SMWDS example this means that the arc capacity is 0, since the indicator is 0 if neither a vertex nor its neighbours are selected. Thus we consider only the highlighted arcs in Figure 3a, which we will denote from left to right a_1 and a_2. For both arcs we have $\pi_u - \pi_v = 1 > 0$, and we only have one indicator expression $x_0 + x_1 + x_3$ on each arc, so we will ignore the indices $i \in \mathcal{I}_a^\omega$ for ease of presentation. We can set $\hat{\beta}_{a_1 i} = \hat{\beta}_{a_2 i} = \pi_u - \pi_v = 1$ and $\hat{\beta}_{ai} = 0$ for all other capacitated arcs a and sets $\mathcal{I}_a^\omega$. We have now obtained the cut:

$$\eta \geq 1 - 2(x_0 + x_1 + x_3) \tag{24}$$

which will be added to the master problem.

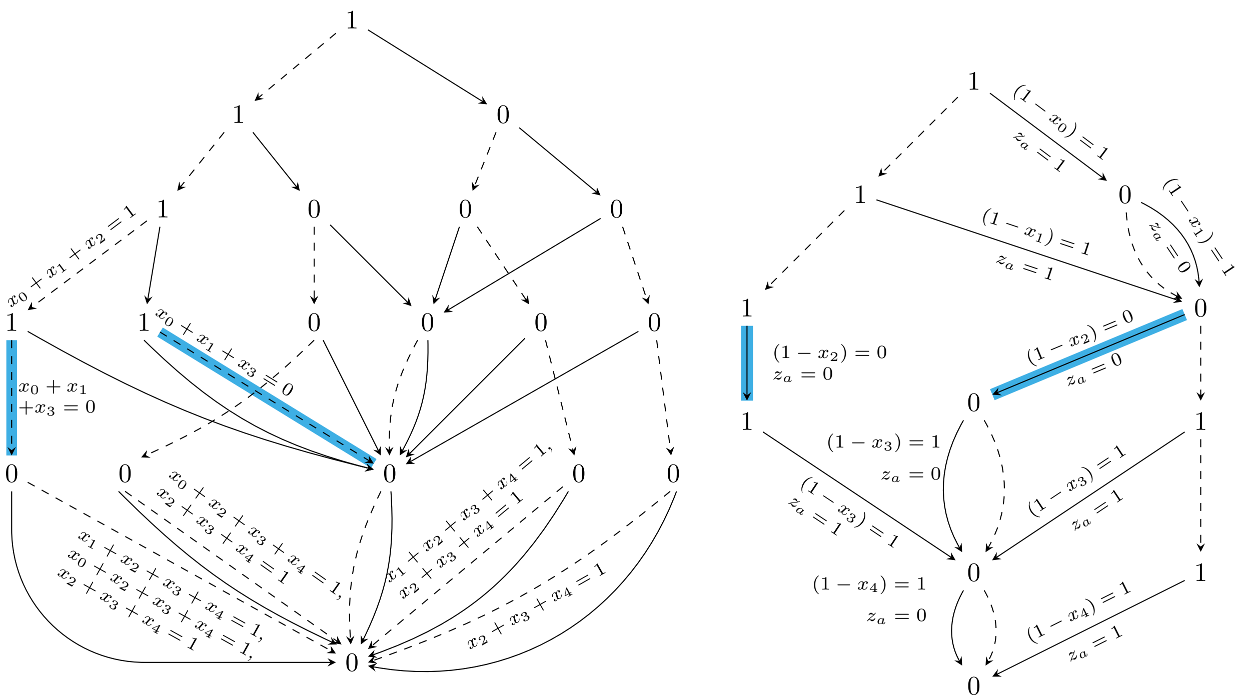
For a fixed scenario ω the Benders cuts that come from `BDD3` are:

$$\eta_\omega \geq \hat{\pi}_{\mathtt{r}} - \sum_{a \in \mathcal{A}^{\mathtt{one},\omega}} \hat{z}_a \mu_{\sigma(a)}^\omega. \tag{25}$$

The procedure to obtain the duals for the `BDD3` subproblems is very similar to that for the `BDD2` subproblems. For a scenario BDD $\mathcal{B}^\omega$ and master problem solution $\hat{\mu}^\omega$, we calculate the dual solutions using a bottom-up shortest-path algorithm, starting at the terminal node which has $\pi_{\mathtt{t}} = 0$. For each subsequent node u, π_u is the length of the shortest u-$\mathtt{t}$ path. As seen in (22), the z_a variables correspond only to arcs in $\mathcal{A}^{\mathtt{one},\omega}$. By constraints (22d) and (22e), if $\pi_u - \pi_v - d^2_{\sigma(a)}(\omega) \leq 0$, we can set $z_a = 0$. Otherwise, we set $z_a = \pi_u - \pi_v - d^2_{\sigma(a)}(\omega)$. In this way, $z_a + d^2_{\sigma(a)}(\omega)$ can be seen as the change in path length between u and v.

Example 4. We will compute a cut at master problem solution $\hat{x} = [0, 0, 1, 0, 0]$ for the BDD built in Example 2. As seen by the highlighted arcs in Figure 3b, at this solution we have two arcs in $\mathcal{A}^{\mathtt{one},\omega}$ that now have cost 0. Using the parametrized arc costs we compute the shortest path dual values π which can be seen inside the nodes in Figure 3b. To find the values of the z_a variables, we will compute $\pi_u - \pi_v - d^2_{\sigma(a)}(\omega)$ for all $a = (u, v) \in \mathcal{A}^{\mathtt{one},\omega}$ and set z_a accordingly. We remark that for this instance we have $d^2_{\sigma(a)}(\omega) = 0$ for all arcs a. These values can be seen next to the arcs in Figure 3b. Taking $x_{\sigma(a)}$ as the indicator we have now obtained the cut:

$$\eta \geq 1 - (x_0 + x_1 + 2x_3 + x_4). \tag{26}$$



(a) `BDD2` subproblem solution for the SMWDS instance in Example 3 at `MP` solution $\hat{x} = [0, 0, 1, 0, 0]$. The nodes contain their dual values, and the highlighted arcs have capacity 0.

(b) `BDD3` subproblem solution for the SMWDS instance in Example 4 at `MP` solution $\hat{x} = [0, 0, 1, 0, 0]$. The nodes contain their dual values, and the highlighted 1-arcs have cost 0.

Figure 3: BDD subproblem solutions for the SMWDS instance in Example 3 and in Example 4.

We remark that this cut differs from the cut found in Example 3 despite being derived for the same subproblem instance at the same master problem solution.

We remark that we are able to apply the cut strengthening strategies proposed by Lozano and Smith [29] to both cuts (23) and (25). The first strategy relies on partitioning $\mathcal{A}^\omega$ into disjoint sets of arcs and selecting the maximal value of $\hat{\beta}_{ai}$ or $\hat{z}_a$ from each set to appear in the cut. First remark that the BDD arc layers, $\mathcal{A}^\omega_j$ for all $j \in \{1, \dots, m_1\}$ are disjoint and partition $\mathcal{A}^\omega$, and that for each $a \in \mathcal{A}$ we must have $\mathcal{I}^\omega_a \subseteq \{1, \dots, m_1\}$. In `BDD2`, for all $j \in \{1, \dots, m_1\}$ and $i \in \{1, \dots, m_1\}$ let $\hat{\beta}^{\max}_{ji} = \max_{a \in \mathcal{A}^\omega_j} \{\hat{\beta}_{ai} : i \in \mathcal{I}^\omega_a\}$. Similarly, in `BDD3`, remark that because we have assumed the natural ordering of the μ^ω variables, for each arc a in a BDD arc layer $\mathcal{A}^\omega_j$, we have $\sigma(a) = j$ for all $j \in \{1, \dots, m_1\}$. In `BDD3`, for all $j \in \{1, \dots, m_1\}$ define $\hat{z}^{\max}_j = \max_{a \in \mathcal{A}^\omega_j} \{\hat{z}_a\}$. Then the cut

$$\eta_\omega \geq \hat{\pi}_{\mathtt{r}} - \sum_{j \in \{1, \dots, m_1\}} \sum_{i \in \{1, \dots, m_1\}} \hat{\beta}^{\max}_{ji} (1 - \mu^\omega_i) \tag{27}$$

is at least as good as (23) and the cut

$$\eta_\omega \geq \hat{\pi}_{\mathtt{r}} - \sum_{j \in \{1,\ldots,m_1\}} \hat{z}_j^{\max} \mu_j^\omega. \tag{28}$$

is at least as good as (25). Henceforth we assume the use of these strengthened cuts for `BDD2` and `BDD3` unless otherwise stated.

The second strengthening strategy solves a secondary optimization problem to help compute lower bounds on the second-stage objective function which are then used to strengthen the cut. We refer the reader to Lozano and Smith [29] for further details on the cut strengthening strategies.

Example 5. For the scenario graph in Figure 1, the strengthened version of the cut (24) is

$$\eta \geq 1 - (x_0 + x_1 + x_3) \tag{29}$$

and the strengthened version of the cut (26) is

$$\eta \geq 1 - (x_0 + x_1 + x_3 + x_4). \tag{30}$$

5.2 Integer L-shaped Method

We present the integer L-shaped method as an alternative decomposition method to the BDD-based decompositions. The integer L-shaped method solves an integer program both in the master problem and in the subproblems. The master problem, `MP`, is the same as for the BDD-based decompositions. The subproblem for scenario ω is the integer program:

$$\min\ d(\omega)^\top y \tag{31a}$$
$$\text{s.t. } y \in \mathcal{Z}(\hat{\mu}^\omega, \omega) \tag{31b}$$
$$y \in \mathcal{Y}(\omega) \tag{31c}$$
$$y \in \{0,1\}^{n_y}. \tag{31d}$$

where $\mathcal{Z}(\hat{\mu}^\omega, \omega)$ are those constraints that depend on the first-stage decisions and correspond to the indicator constraints for the BDD-based methods. Since we have binary variables in the subproblems, we use logic-based Benders cuts. Given a scenario ω and a master problem solution $\hat{\mu}^\omega$, let τ_ω be the optimal objective value of (31). The standard cuts are called "no-good" cuts and take the following form:

$$\eta_\omega \geq \tau_\omega - \tau_\omega \left(\sum_{\substack{i \in \{1,\ldots,m_1\} \\ \hat{\mu}_i^\omega = 1}} (1 - \mu_i^\omega) + \sum_{\substack{i \in \{1,\ldots,m_1\} \\ \hat{\mu}_i^\omega = 0}} \mu_i^\omega \right). \tag{32}$$

We remark that based on problem-specific structure, these cuts can be strengthened. We demonstrate this in Section 7 on the SMWDS problem.

5.3 Pure Benders Cuts

In order to strengthen any of the decomposition methods, we have incorporated a class of cuts based on the solution of the LP relaxation of (31) which we call pure Benders (PB) cuts. We pass a solution from `MP` to (31) and solve without integrality constraints. We then use the duals of this solution to add to `MP` a traditional Benders cut of the form

$$\eta_\omega \geq \mathcal{T}_{\hat{\mu}^\omega}(\mu^\omega), \tag{33}$$

where the function $\mathcal{T}_{\hat{\mu}^\omega}(\mu^\omega)$ returns an affine expression of μ^ω.

5.4 Cut Strength Comparison

In this section we compare the strength of the proposed cuts, starting with the strength of the integer L-shaped cuts as compared to the BDD-based cuts. Next we compare the two types of BDD cuts and finally we examine the strength of the PB cuts.

We remark that the following Propositions 4 and 5 rely on non-negative objective coefficients in the second-stage problems, which is quite common in applications as we are often minimizing costs.

Proposition 4. *Given a scenario ω and a master problem solution $\hat{\mu}^\omega$, if the subproblem objective coefficients are non-negative, i.e., $d_i(\omega) \geq 0$ for all $i \in \{1, \ldots, n_y\}$, then the strengthened `BDD2` cuts (27) are at least as good as the integer L-shaped cuts (32).*

Proof. Assume $d_i(\omega) \geq 0$ for all $i \in \{1, \ldots, n_{\mathtt{y}}\}$. We want to show the strengthened `BDD2` cuts (27) are as good as integer L-shaped cuts (32) for all $i \in \{1, \ldots, n_{\mathtt{y}}\}$, that is,

$$\hat{\pi}_{\mathtt{r}} - \sum_{j \in \{1,\ldots,m_1\}} \sum_{i \in \{1,\ldots,m_1\}} \hat{\beta}^{\max}_{ji}(1 - \mu^\omega_i) \geq \tau_\omega - \tau_\omega \left(\sum_{\substack{i \in \{1,\ldots,m_1\} \\ \hat{\mu}^\omega_i = 1}} (1 - \mu^\omega_i) + \sum_{\substack{i \in \{1,\ldots,m_1\} \\ \hat{\mu}^\omega_i = 0}} \mu^\omega_i \right).$$

As described in Section 5.1, for each capacitated arc a, we have $\hat{\beta}_{ai} > 0$ for exactly one $i \in \mathcal{I}^\omega_a$ where $\hat{\mu}^\omega_i = 1$ and otherwise $\hat{\beta}_{ai} = 0$. Therefore if $\hat{\beta}^{\max}_{ji} > 0$, we must have $(1 - \hat{\mu}^\omega_i) = 0$, and otherwise $\hat{\beta}^{\max}_{ji} = 0$, since $\hat{\beta}^{\max}_{ji} = \max_{a \in \mathcal{A}^\omega_j}\{\hat{\beta}_{ai} : i \in \mathcal{I}^\omega_a\}$. As a result for all $j \in \{1, \ldots, m_1\}$ and $i \in \{1, \ldots, m_1\}$, we have $\hat{\beta}^{\max}_{ji}(1 - \hat{\mu}^\omega_i) = 0$ at an optimal solution. By Proposition 1, τ_ω is the optimal objective value to an equivalent subproblem to that of `BDD2`, so we have

$$\hat{\pi}_{\mathtt{r}} - \sum_{j \in \{1,\ldots,m_1\}} \sum_{i \in \{1,\ldots,m_1\}} \hat{\beta}^{\max}_{ji}(1 - \hat{\mu}^\omega_i) = \hat{\pi}_{\mathtt{r}} = \tau_\omega.$$

Substituting $\hat{\pi}_{\mathtt{r}}$ for τ_ω, and swapping the summation indices, we would like to show

$$\sum_{i \in \{1,\ldots,m_1\}} \sum_{j \in \{1,\ldots,m_1\}} \hat{\beta}^{\max}_{ji}(1 - \mu^\omega_i) \leq \hat{\pi}_{\mathtt{r}} \left(\sum_{\substack{i \in \{1,\ldots,m_1\} \\ \hat{\mu}^\omega_i = 1}} (1 - \mu^\omega_i) + \sum_{\substack{i \in \{1,\ldots,m_1\} \\ \hat{\mu}^\omega_i = 0}} \mu^\omega_i \right). \tag{34}$$

By definition of the strengthened cuts (27), we select at most one arc $a = (u, v)$ per BDD layer j and index i. By construction of the solution, $\hat{\beta}_{ai} > 0$ for the index $i \in \mathcal{I}^\omega_a$ only when $\hat{\mu}^\omega_i = 1$. Therefore we also have $\hat{\beta}^{\max}_{ji} > 0$ only when $\hat{\mu}^\omega_i = 1$. Thus the outer left-hand term of (34) and $\sum_{\substack{i \in \{1,\ldots,m_1\} \\ \hat{\mu}^\omega_i = 1}} (1 - \mu^\omega_i)$ sum over the same μ^ω indices.

Without loss of generality, fix $i = 1$, for each BDD arc layer, $\mathcal{A}^\omega_j$, there exits $\bar{a} \in \mathcal{A}^\omega_j$ such that $\hat{\beta}_{\bar{a}1} = \hat{\beta}^{\max}_{j1}$. The value of $\hat{\beta}_{\bar{a}1} + d_{\sigma(\bar{a})}(\omega) = \pi_{\bar{u}} - \pi_{\bar{v}}$ is the path length increase over the arc $\bar{a}$ since $d_{\sigma(a)}(\omega) \geq 0$ for all $a \in \mathcal{A}$. By construction of the cut, we will select at most one arc $\bar{a}$ per layer $\mathcal{A}^\omega_j$, and $\hat{\beta}_{\bar{a}1} > 0$ only if $\bar{a}$ is on the shortest path and $\hat{\mu}^\omega_1 = 1$. Thus, $\hat{\beta}^{\max}_{j1} + d_{\bar{a}}(\omega)$ is the path length increase over layer j. Since at most every violated index will exist on every layer, and $\hat{\pi}_{\mathtt{r}}$ is the total path length increase over the entire BDD, we have

$$\sum_{j \in \{1,\ldots,m_1\}} \hat{\beta}^{\max}_{j1} + \sum_{a \in \bar{A}} d_a(\omega) \leq \hat{\pi}_{\mathtt{r}},$$

where $\bar{A}$ is a set of one arc a per layer $\mathcal{A}^\omega_j$ such $\hat{\beta}_{a1} = \hat{\beta}^{\max}_{j1}$. We have $d_{\sigma(a)}(\omega) \geq 0$ by assumption, thus $\sum_{j \in \{1,\ldots,m_1\}} \hat{\beta}^{\max}_{ji} \leq \hat{\pi}_{\mathtt{r}}$ for any i. Therefore we have

$$\sum_{i \in \{1,\ldots,m_1\}} \sum_{j \in \{1,\ldots,m_1\}} \hat{\beta}^{\max}_{ji}(1 - \mu^\omega_i) \leq \sum_{\substack{i \in \{1,\ldots,m_1\} \\ \hat{\mu}^\omega_i = 1}} \hat{\pi}_{\mathtt{r}}(1 - \mu^\omega_i).$$

By definition $1 \geq \mu^\omega_i \geq 0$, and by construction of dual solutions and since $d_i(\omega) \geq 0$ by assumption, $\hat{\pi}_{\mathtt{r}} \geq 0$ thus

$$\sum_{\substack{i \in \{1,\ldots,m_1\} \\ \hat{\mu}^\omega_i = 1}} \hat{\pi}_{\mathtt{r}}(1 - \mu^\omega_i) \leq \hat{\pi}_{\mathtt{r}} \left(\sum_{\substack{i \in \{1,\ldots,m_1\} \\ \hat{\mu}^\omega_i = 1}} (1 - \mu^\omega_i) + \sum_{\substack{i \in \{1,\ldots,m_1\} \\ \hat{\mu}^\omega_i = 0}} \mu^\omega_i \right).$$

Thus by transitivity the equation (34) holds and we conclude the cuts (27) are at least as good as the cuts (32) when $d_i(\omega) \geq 0$ for all $i \in \{1, \ldots, n_{\mathtt{y}}\}$. □

Proposition 5. *Given a scenario ω and a master problem solution $\hat{\mu}^\omega$, if the subproblem objective coefficients are non-negative, i.e., $d^2_i(\omega) \geq 0$ for all $i \in \{1, \ldots, n_{\mathtt{y}}\}$, then the strengthened `BDD3` cuts (28) are at least as good as the integer L-shaped cuts (32).*

Proof. Assume $d^2_i(\omega) \geq 0$ for all $i \in \{1, \ldots, n_{\mathtt{y}}\}$ and observe that by Proposition 2, we have $\tau_\omega = \hat{\pi}_{\mathtt{r}}$ at an optimal subproblem solution. Thus we would like to show:

$$\sum_{j \in \{1,\ldots,m_1\}} \hat{z}^{\max}_j \mu^\omega_j \leq \hat{\pi}_{\mathtt{r}} \left(\sum_{\substack{i \in \{1,\ldots,m_1\} \\ \hat{\mu}^\omega_i = 1}} (1 - \mu^\omega_i) + \sum_{\substack{i \in \{1,\ldots,m_1\} \\ \hat{\mu}^\omega_i = 0}} \mu^\omega_i \right). \tag{35}$$

We remark that at an optimal subproblem solution, $\hat{z}_a + d^2_{\sigma(a)}(\omega)$ can be seen as the change in path length over arc a. We have $d^2_i(\omega) \geq 0$ for all $i \in \{1, \ldots, n_{\mathtt{y}}\}$ by assumption and $\hat{z}_a \geq 0$ by definition (22e), therefore $\hat{z}_a + d^2_{\sigma(a)}(\omega) \geq 0$ for all $a \in \mathcal{A}^\omega$ and the path length change over every arc in the BDD is non-negative. Thus we must have $0 \leq \hat{z}_a + d^2_{\sigma(a)}(\omega) \leq \hat{\pi}_{\mathtt{r}}$ since by definition of the cuts $z_a + d^2_{\sigma(a)}(\omega)$ is the path length increase over a single layer and $\hat{\pi}_{\mathtt{r}}$ is the total path length. Therefore, since for all $j \in \{1, \ldots, m_1\}$ there exists an arc a where $\hat{z}^{\max}_j = \hat{z}_a$, and $d^2_j(\omega) \geq 0$ by assumption, we must also have $\hat{z}^{\max}_j \leq \hat{\pi}_{\mathtt{r}}$.

By definition of the strengthened cuts (28) we know there is exactly one $\hat{z}_a$ term per layer in the cut, i.e., $\hat{z}^{\max}_j$. We also know that if $\hat{\mu}^\omega_{\sigma(a)} = 1$ the cost of all arcs in a layer j, are 0, meaning

$\hat{z}_a = 0$ for all $a \in \mathcal{A}_j^\omega$ as well. Thus the only terms which appear in the BDD3 cut are those where $\hat{\mu}^\omega_{\sigma(a)} = 0$, which allow $\hat{z}_a \geq 0$, and therefore $\hat{z}_j^{\max} \geq 0$. Thus the summation on the left-hand side of (35) is over the same terms as the summation on the right-hand side with $\hat{\mu}^\omega_{\sigma(a)} = 0$. Thus with $\hat{z}_j^{\max} \leq \hat{\pi}_{\mathtt{r}}$ we have

$$\sum_{j\in\{1,\ldots,m_1\}} \hat{z}_j^{\max}\mu_j^\omega \leq \sum_{\substack{i\in\{1,\ldots,m_1\}\\ \hat{\mu}_i^\omega=0}} \hat{\pi}_{\mathtt{r}}\mu_i^\omega.$$

By definition $1 \geq \mu_i^\omega \geq 0$, for all $i \in \{1,\ldots,m_1\}$ and we have shown $\hat{\pi}_{\mathtt{r}} \geq 0$ thus

$$\sum_{\substack{i\in\{1,\ldots,m_1\}\\ \hat{\mu}_i^\omega=0}} \hat{\pi}_{\mathtt{r}}\mu_i^\omega \leq \sum_{\substack{i\in\{1,\ldots,m_1\}\\ \hat{\mu}_i^\omega=1}} \hat{\pi}_{\mathtt{r}}(1-\mu_i^\omega) + \sum_{\substack{i\in\{1,\ldots,m_1\}\\ \hat{\mu}_i^\omega=0}} \hat{\pi}_{\mathtt{r}}\mu_i^\omega$$

and by transitivity we have (35). Thus we conclude if $d_i^2(\omega) \geq 0$ for all $i \in \{1,\ldots,n_{\mathtt{y}}\}$ then the cuts (28) are at least as good as the cuts (32).

□

Next we show the two kinds of BDD cuts and the PB cuts are incomparable. To do so we derive cuts at two master problem solutions using the SMWDS scenario in Figure 1. First we fix MP solution $\hat{x} = [0,0,0,0,0]$ and derive the following cuts:

$$\mathtt{BDD2} : \eta \geq 2 - (2x_0 + 3x_1 + 3x_2 + 3x_3 + 2x_4) \tag{36}$$

$$\mathtt{BDD3} : \eta \geq 2 - (x_0 + x_1 + x_2 + x_3 + x_4) \tag{37}$$

$$\text{PB} : \eta \geq 1.5 - (x_0 + x_1 + x_2 + x_3 + 0.5x_4). \tag{38}$$

Next we fix MP solution $\hat{x} = [0,0,1,0,0]$ and derive the cuts from Example 5, (29) and (30), and the following PB cut:

$$\text{PB} : \eta \geq 1 - (x_0 + x_1 + x_3). \tag{39}$$

In the following propositions we compare the strength of the derived cuts.

Proposition 6. *The strengthened BDD2 cuts (27) are incomparable with the pure Benders cuts (33).*

Proof. First we show a point where (27) dominates (33). Let $\hat{x} = [0.25,0,0,0,0]$ and let $\hat{\eta} = 1.3$. The inequality (36) becomes $\eta \geq 1.5$ which will cut off the current solution of $\hat{\eta} = 1.3$, however the inequality (38) becomes $\eta \geq 1.25$ which does not cut off the current solution. Next we show a point where (33) dominates (27), let $\hat{x} = [0,0,0,0,1]$ and let $\hat{\eta} = 0.5$. The inequality (36) becomes $\eta \geq 0$ which does not cut off the current solution. The inequality (38) becomes $\eta \geq 1$ which cuts off the solution as $\hat{\eta} = 0.5$. We conclude these cuts are incomparable. □

Proposition 7. *The strengthened BDD3 cuts (28) are incomparable with the pure Benders cuts (33).*

Proof. Let $\hat{x} = [1,0,0,0,0]$ and let $\hat{\eta} = 0.75$, then the inequality (37) becomes $\eta \geq 1$ and (38) becomes $\eta \geq 0.5$ Thus (37) cuts off the current solution $\hat{\eta} = 0.75$ and (38) does not. Let $\hat{x} = [0,0,0,0,1]$ and let $\hat{\eta} = 0.5$, the inequality (30) becomes $\eta \geq 0$ which does not cut off the current solution and (39) becomes $\eta \geq 1$ which does. We conclude these cuts are incomparable. □

Proposition 8. *The strengthened BDD2 cuts (27) and the strengthened BDD3 cuts (28) are incomparable.*

Proof. Let $\hat{x} = [0, 0, 0, 0, 1]$ and let $\hat{\eta} = 0.5$, the inequality (36) becomes $\eta \geq 0$ which does not cut off the solution $\hat{\eta} = 0.5$. The inequality (37) becomes $\eta \geq 1$ which cuts off the current solution. Keeping the same $\hat{x}$ and $\hat{\eta}$ solution, the inequality (29) becomes $\eta \geq 1$ which cuts off $\hat{\eta} = 0.5$. However, the inequality (30) becomes $\eta \geq 0$ which does not cut off the current solution. Thus we conclude these cuts are incomparable. □

Propositions 6, 7, and 8 indicate that it may be beneficial to add cuts from multiple classes at each iteration. We will compare the strength of the cuts in practice on the SMWDS application in Section 7. In the following section we extend the two-stage problem to a risk-averse setting.

6 Incorporation of Conditional Value at Risk

Incorporating risk measures into stochastic programs is a well-established practice in the literature, however there is limited work when the second-stage problem has integer variables [42, 53]. To our knowledge, we propose the first decomposition method for 2SP with the conditional value-at-risk (CVaR) risk measure and integer second-stage variables. We will use BDDs or integer L-shaped cuts to convexify the second-stage value function. We remark that this decomposition is rooted in the work of Noyan [31] who considers a decomposition approach for risk-averse 2SP with *continuous* second-stage variables. We will consider the two-stage mean-risk stochastic program

$$\min_{x \in \mathcal{X}} \ \mathbb{E}[g(x,\omega)] + \lambda CVaR_\alpha(g(x,\omega)). \tag{40}$$

where $g(x,\omega) = c^\top x + \mathcal{Q}(x,\omega)$ is the total cost function of the first-stage problem for a fixed scenario, $\lambda \geq 0$ is a risk coefficient, and where $CVaR_\alpha$ is the CVaR at level $\alpha \in (0,1]$.

Formally, the value-at-risk (VaR) of a random variable Z at level α is the α-quantile

$$VaR_\alpha(Z) = \inf_{\zeta \in \mathbb{R}} \{\zeta : \mathbb{P}[Z \leq \zeta] \geq \alpha\}. \tag{41}$$

VaR_α is the smallest value ζ such that with probability α the random variable Z will not exceed ζ [38]. Thus VaR_α gives an upper bound on the random variable Z that will only be exceeded with a probability of $(1-\alpha)$. However, VaR does not quantify the severity of exceeding this upper bound. This is why CVaR is a useful risk measure.

CVaR, also called expected shortfall, is the conditional expectation that the random variable Z exceeds VaR_α

$$CVaR_\alpha(Z) = \mathbb{E}[Z : Z \geq VaR_\alpha(Z)]. \tag{42}$$

Thus it measures the severity of exceeding VaR_α. More formally, letting $[a]_+ = \max\{0, a\}$ for $a \in \mathbb{R}$, CVaR of a random variable Z at confidence level $\alpha \in (0,1]$ is defined as

$$CVaR_\alpha(Z) = \inf_{\zeta \in \mathbb{R}} \left\{ \zeta + \frac{1}{1-\alpha} \mathbb{E}([Z - \zeta]_+) \right\}. \tag{43}$$

Applying the translational invariance of $CVaR_\alpha$, the mean-risk version of problem (5) at confidence level α is

$$\min_{x \in \mathcal{X}} \ (1+\lambda)c^\top x + \mathbb{E}[\mathcal{Q}(x,\omega)] + \lambda CVaR_\alpha(\mathcal{Q}(x,\omega)). \tag{44}$$

Thus we take the $CVaR_\alpha$ of the recourse function $\mathcal{Q}(x,\omega)$, substituting its objective function for the random variable Z. For a finite set of scenarios Ω with probabilities p_ω for $\omega \in \Omega$, and with the second-stage problem (6), this yields the following monolithic mixed-integer program

$$
\begin{aligned}
\min\ & (1+\lambda)c^\top x + \sum_{\omega\in\Omega} p_\omega (d^1(\omega)+d^2(\omega))^\top y^\omega + \lambda\left(\zeta + \frac{1}{1-\alpha}\sum_{\omega\in\Omega} p_\omega \nu^\omega\right) && && \text{(45a)}\\
\text{s.t. } & x \in \mathcal{X} && && \text{(45b)}\\
& \mathbb{I}\left(L_i^\omega(x)\right) = 1 \implies d^1_{q(i)}(\omega) = 0 && \forall i = 1,\ldots,m_1, \forall \omega\in\Omega && \text{(45c)}\\
& y^\omega \in \mathcal{Y}(\omega) && \forall \omega \in \Omega && \text{(45d)}\\
& \nu^\omega \geq (d^1(\omega)+d^2(\omega))^\top y^\omega - \zeta && \forall \omega \in \Omega && \text{(45e)}\\
& \nu^\omega \geq 0 && \forall \omega \in \Omega && \text{(45f)}\\
& y^\omega \in \{0,1\}^{n_y} && \forall \omega \in \Omega && \text{(45g)}\\
& \zeta \in \mathbb{R} && && \text{(45h)}
\end{aligned}
$$

In the decomposition approaches presented by Noyan [31], the second-stage problem is a linear program, thus convexity of $\mathcal{Q}(x,\omega)$ in x for all $\omega \in \Omega$ follows easily, and they use linear programming duality to approximate the value of the mean-risk function of the recourse cost. The key to our decomposition is the observation that we have already convexified the approximation of the recourse problem using the two versions of BDD-based cuts and the integer L-shaped cuts. We will now present a decomposition algorithm that solves (45) using our previously derived cuts. For ease of presentation we will show the decomposition using the procedure `BDD3` and corresponding cuts (28), but remark that use of the procedure `BDD2` or of integer L-shaped cuts is equally valid.

6.1 Decomposition Algorithm

In this algorithm, we solve the second-stage problems and generate cuts from the scenario BDDs as in Section 5. We will take the optimal recourse function value from the BDD subproblem and use it in the $CVaR_\alpha$ expression. For example, using the `BDD3` reformulation, the CVaR value of the recourse problem for a single scenario becomes

$$\inf_{\zeta\in\mathbb{R}} \left\{ \zeta + \frac{1}{1-\alpha}\left[\left(\hat{\pi}_{\mathtt{r}} - \sum_{j\in\{1,\ldots,m_1\}} \hat{z}_j^{\max}\mu_j^\omega\right) - \zeta\right]_+ \right\}. \tag{46}$$

In the master problem of this decomposition, as in `MP`, we introduce variables η_ω to model the objective value of the recourse problem and we introduce new variables θ_ω to model the value of CVaR for each scenario $\omega \in \Omega$. We remark that we can now replace the optimal recourse cost with η_ω in (46), since at any optimal solution we will have $\eta_\omega = \hat{\pi}_{\mathtt{r}}^k - \sum_{j\in\{1,\ldots,m_1\}} \hat{z}_j^{\max k}\mu_j^\omega$ for the k^{th} cut generated. This is in contrast to the work of Noyan [31], who use the original optimal recourse cost in their model. We also define variables ζ^k to model VaR_α of the recourse cost and $\nu^{\omega k}$ to model the function $[\eta_\omega - \zeta^k]_+, \forall \omega \in \Omega$ in cut k.

We can now formulate the master problem after K cuts have been generated as

$$\min\ (1+\lambda)c^\top x + \sum_{\omega\in\Omega} p_\omega \eta_\omega + \lambda \sum_{\omega\in\Omega} p_\omega \theta_\omega \tag{47a}$$

$$
\begin{aligned}
\text{s.t. } & x \in \mathcal{X} && && \text{(47b)}\\
& \mathbb{I}(L_i^\omega(x)) = 1 \implies \mu_i^\omega = 1 && \forall i = 1, \dots, m_1 && \text{(47c)}\\
& \eta_\omega \geq \hat{\pi}_{\mathtt{r}}^k - \sum_{j \in \{1,\dots,m_1\}} \hat{z}_j^{\max k} \mu_j^\omega && \forall \omega \in \Omega, k = 1, \dots, K && \text{(47d)}\\
& \theta_\omega \geq \zeta^k + \frac{1}{1-\alpha} \nu^{\omega k} && \forall \omega \in \Omega, k = 1, \dots, K && \text{(47e)}\\
& \nu^{\omega k} \geq \eta_\omega - \zeta^k && \forall \omega \in \Omega, k = 1, \dots, K && \text{(47f)}\\
& \nu^{\omega k} \geq 0 && \forall \omega \in \Omega, k = 1, \dots, K && \text{(47g)}\\
& \zeta^k \in \mathbb{R} && k = 1, \dots, K && \text{(47h)}
\end{aligned}
$$

where the optimality cuts (47d) are generated using the `BDD3` procedure. These cuts could be replaced with those from the `BDD2` procedure or with integer L-shaped cuts derived from the original second-stage problem.

6.1.1 Algorithm Overview

0. Initialize: Set iteration counter $k = 0$ and set the upper bound $UB_0 = \infty$. If using the BDD-based methods, generate a BDD $\mathcal{B}^\omega$ for each scenario $\omega \in \Omega$ using either `BDD2` or `BDD3`.

1. Solve the master problem (47) and get optimal solution $(\hat{x}, \hat{\mu}, \hat{\eta}, \hat{\theta})$. Set LB_k to the objective value at this solution.

2. Solve each subproblem for the dual solution and yielding objective value τ_ω. Set $k = k + 1$.

3. For each $\omega \in \Omega$, if $\tau_\omega > \hat{\eta}_\omega$ add the optimality cut (47d) using the subproblem solution.

4. Calculate the α-quantile, ζ, of the set of recourse problem realizations, and then calculate

$$CVaR_\alpha(\tau_\omega, \zeta) = \zeta + \frac{1}{1-\alpha}[\tau_\omega - \zeta]_+.$$

5. For each $\omega \in \Omega$, if $CVaR_\alpha(\tau_\omega, \zeta) > \hat{\theta}_\omega$ add the CVaR optimality cuts (47e), (47f), (47g), (47h) using the subproblem solution and introducing axillary variables.

6. If neither Step 3 nor Step 5 yield cuts, accept the incumbent as an optimal solution and terminate. Otherwise, go to Step 1.

7 Computational Results on a Novel Application

7.1 The Stochastic Minimum Weight Dominating Set Problem

In the design of wireless sensor networks it is often useful to partition the network so that a subset of sensors are more powerful than the others and are able coordinate the rest of the network [39]. Ideally, these powerful sensors will be linked to every other sensor in the network which amounts to finding a dominating set in the wireless sensor network [7]. Finding a dominating set of size less than a given number is a classical NP-complete decision problem [17]. In practice we often wish to identify the Minimum Weight Dominating Set (MWDS), where node weights usually represent mobility characteristics or characteristics such as the residual energy or the available bandwidth [8]. Other applications of MWDS include disease suppression and control [51], the study of public goods in networks [9], social network analysis [28], and routing in ad hoc wireless networks [56].

In many applications, the network structure is uncertain or time dependent, thus a deterministic setting is not a realistic approximation of this problem [2, 14, 28, 50]. We extend this problem to the stochastic setting, where the nodes of the graph can fail and where the node weights can change, which will help capture the inherent uncertainty in the problem structure.

Work on the MWDS problem in the stochastic setting has been limited. In [7], a graph with an existing MWDS is given as input and the nodes of the graph fail with some given probability, the goal is then to find a new dominating set which includes any functional sensors from the original solution. This reflects the real-world case of a portion of the network being damaged and needing repair or replacement. The authors propose algorithms to solve this problem in cases where the problem is polynomial and propose polynomial approximations for general graphs. Finally, they give an IP formulation for this problem but do not implement it. In [52], the graph is fixed but the node weights are stochastic, and the objective is to find a connected dominating set. The probability distribution function of the random weights is unknown. They present a series of learning automata-based algorithms to solve this problem and show they reduce the number of samples needed to construct a solution. A third variant of the MWDS in the probabilistic setting is presented by [20], where the graph edges can transmit signals with a given probability and the objective is to find a connected dominating set with a certain reliability. Similar stochastic problems have been studied in wildfire evacuation planning [14] power system monitoring [50].

In the probabilistic MWDS problem on stochastic graphs (SMWDS) we assume the existence of a graph where the vertex weights are known and no failure has been observed yet. For example, when a network has been constructed but is not yet operational. We then observe some change in the network, wherein the weights may increase or decrease and any vertex may fail and be removed from the graph. For example, when the network is operational and experiences equipment failures and changes in operating costs. To the best of our knowledge, we present the first study of the probabilistic MWDS problem on stochastic graphs.

Given a graph $G = (\mathcal{V}, \mathcal{E})$, let $c_v > 0 \ \forall \ v \in \mathcal{V}$ be the first-stage vertex weights and let $d_v(\omega) \geq 0 \ \forall \ v \in \mathcal{V}$ be the second-stage vertex-weight random variables. Define binary decision variables $x_v = 1$ if the vertex $v \in \mathcal{V}$ is assigned to the dominating set in the first stage, and $y_v = 1$ if the vertex v is assigned to the dominating set in the second stage under equally probable scenario $\omega \in \Omega$. We also define $\mathcal{N}(v) = \{u \in \mathcal{V} : \{u, v\} \in \mathcal{E}\}$, $\mathcal{N}'_\omega(v) = \{u \in \mathcal{V} : \{u, v\} \in \mathcal{E}, d_u(\omega) \neq 0\}$ and $\mathcal{V}'_\omega = \{v \in \mathcal{V} : d_v(\omega) \neq 0\}$. The SMWDS problem is formulated as follows

$$\min \sum_{v \in \mathcal{V}} c_v x_v + \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \mathcal{Q}(x, \omega) \tag{48a}$$

$$\text{s.t. } x_v + \sum_{u \in \mathcal{N}(v)} x_u \geq 1 \qquad \forall \ v \in \mathcal{V} \tag{48b}$$

$$x_v \in \{0, 1\} \qquad \forall \ v \in \mathcal{V} \tag{48c}$$

where

$$\mathcal{Q}(x, \omega) = \min \sum_{v \in \mathcal{V}} d_v(\omega) y_v \tag{49a}$$

$$\text{s.t. } y_v + \sum_{u \in \mathcal{N}'_\omega(v)} y_u \geq 1 - \left(\hat{x}_v + \sum_{u \in \mathcal{N}'_\omega(v)} \hat{x}_u \right) \quad \forall\, v \in \mathcal{V}'_\omega \tag{49b}$$

$$y_v \in \{0, 1\} \quad \forall\, v \in \mathcal{V}'_\omega. \tag{49c}$$

Given this formulation, the constraints (6b) become

$$x_v = 1 \implies d_v(\omega) = 0, \forall\, v \in \mathcal{V}'_\omega, \omega \in \Omega \tag{50}$$

which clearly do not need indicator variables, and the constraints (4b) become

$$x_v + \sum_{u \in \mathcal{N}'_\omega(v)} x_u = 0 \implies y_v + \sum_{u \in \mathcal{N}'_\omega(v)} y_u \geq 1, \forall\, v \in \mathcal{V}'_\omega, \omega \in \Omega. \tag{51}$$

Constraints (51) also do not need indicator variables as we will set $\hat{\mu}_v^\omega = x_v + \sum_{u \in \mathcal{N}'_\omega(v)} x_u$ in the reformulation. This ensures that if a vertex or its neighbour is selected in the first stage and that vertex exists in the scenario, then the cover constraints (49b) become redundant.

We build the subproblem BDDs as in Examples 1 and 2, and the cuts for the BDD-based algorithms are exactly as seen in (27) and (28) replacing the variables μ^ω with the respective indicator expressions. We will also compare the BDD-based algorithms with the integer L-shaped approach, where we solve the subproblems (49) using integer programming. The strengthened integer L-shaped cuts (32) for this problem are:

$$\eta_\omega \geq \mathcal{Q}(\hat{x}, \omega) \left(1 - \sum_{\substack{v \in \mathcal{V} \\ \hat{x}_v = 0}} x_v \right). \tag{52}$$

Finally, we also compare the solution methods with and without the PB cuts which are given by relaxing the binary constraints, removing the upper bounds on the y variables, and solving (49) as a linear program. We then take traditional Benders cut from the dual solution. Letting $\hat{\delta}$ be the optimal dual variables associated with the constraints (49b), then the PB optimality cuts are:

$$\eta_\omega \geq \sum_{v \in \mathcal{V}'_\omega} \hat{\delta}_v \left(1 - x_v - \sum_{u \in \mathcal{N}'_\omega(v)} x_u \right). \tag{53}$$

Given these solution methods, the extension to the risk averse setting is natural, using (47) as the master problem and adding the appropriate CVaR and optimality cuts given the solution method. In the next section we compare these methods on SMWDS instances.

7.2 Computational Results

We compare both BDD-based decomposition algorithms against the integer L-shaped method, with and without PB cuts. We implement the algorithms in C++ and use CPLEX 12.10 to solve, the experiments were conducted on a single thread on a Linux workstation with 3.6GHz Intel Core i9-9900K CPUs and 40GB RAM with a solution time limit of 3600 seconds after the BDDs are generated. We implement the decomposition algorithms as a branch-and-cut procedure, using

CPLEX lazy callbacks to add optimality cuts and PB cuts at integer feasible solutions. We note that PB cuts could also be added at all feasible solutions. Finally, we use CPLEX heuristic callbacks, which are called at every LP feasible solution that also has a better relaxation value than the best available integer solution. The callback then constructs an integer feasible solution to use as a new incumbent.

The instance set is comprised of 200 randomly generated instances based on benchmarks from deterministic MWDS literature [23]. These instances have $|\mathcal{V}| \in \{30, 35, ..., 50\}$ and edge density $\in \{0.2, 0.4, 0.6, 0.8\}$. We conduct sample average approximation analysis and conclude that to achieve a worst-case optimality gap below 1%, we must use at least 800 scenarios and proceed to use 850 scenarios in our experiments. Table 1 gives the 95% confidence interval (CI) on the upper and lower bounds, respectively.

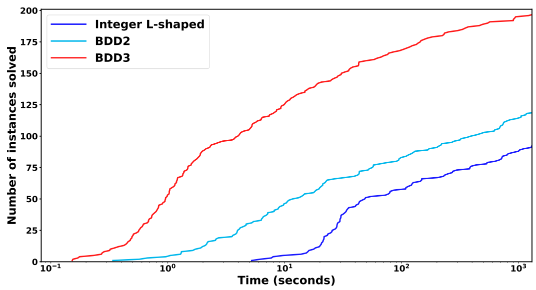
Table 1: SAA anaylsis for an instance with 30 vertices and edge density 0.5.

$\lvert\Omega\rvert$	95% CI on Lower Bound	95% CI on Upper Bound	Worst Case Optimality Gap (%)
10	[34.147, 38.103]	[37.574, 37.634]	10.21
25	[35.509, 38.879]	[37.569, 37.629]	5.97
50	[35.761, 38.249]	[37.590, 37.650]	5.28
100	[36.666, 38.375]	[37.538, 37.591]	2.52
250	[36.789, 37.671]	[36.988, 37.040]	0.68
500	[36.731, 37.477]	[37.181, 37.236]	1.37
600	[36.664, 37.330]	[37.259, 37.301]	1.74
750	[36.810, 37.340]	[37.143, 37.211]	1.09
800	[36.846, 37.367]	[37.131, 37.205]	0.98
850	[36.865, 37.346]	[37.123, 37.183]	0.86

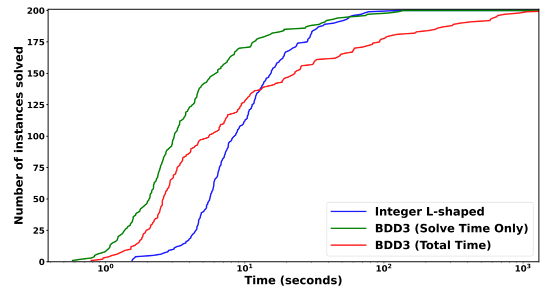
We first compare both BDD-based decomposition methods against the integer L-shaped method without PB cuts. As seen in Figure 4a, which shows the total time (taking into account the building of the BDDs) the `BDD3` method of (6) performs best and is able to solve all instances within the time limit, while the integer L-shaped algorithm is only able to solve 103 of the 200 instances within the time limit. The `BDD2` method is able to solve 135 instances, however it hits the memory limit on 28 instances while building the BDDs, for this reason we exclude it from the remainder of our study. Results with PB cuts can be seen in Figure 4b. The PB cuts for this problem are very strong and allow the integer L-shaped algorithm to perform best on the instances with low density. The `BDD3` method outperforms integer L-shaped on 135 instances, and we can see that the main obstacle for the remaining 35 instances is the time it takes to generate the BDDs for the scenario subproblems.

Next, we compare the algorithms in the CVaR setting with $\lambda = 0.1$ and $\alpha = 0.9$, and separating the CVaR cuts only after there are no remaining recourse optimality cuts, and using PB cuts. These results can be seen in Figure 5, where the `BDD3` method performs best on all but 5 instances.

Overall, we conclude that the `BDD3` method performs best on this application problem, as it requires less memory to build the BDDs than the `BDD2` method and is better than or comparable to the integer L-shaped method with PB cuts on the standard and risk-averse formulations.



(a) Comparison of methods without PB cuts.



(b) Comparison of methods with PB cuts.

Figure 4: Results figures, where we denote the methods using the BDD reformulation of (4) and of (6) as `BDD2` and `BDD3`, respectively.

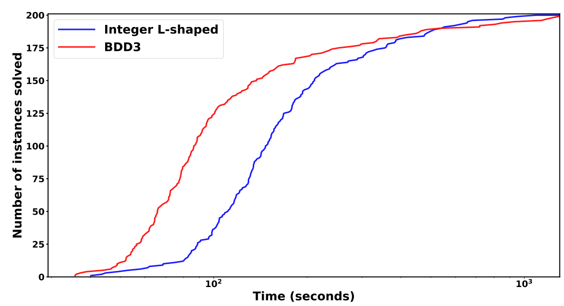


Figure 5: Comparison of methods using PB cuts on CVaR with $\lambda = 0.1$ and $\alpha = 0.9$.

8 Conclusion

In this paper, we considered a class of two-stage stochastic programming problems with binary second-stage variables and characterized by implication constraints, which cover a wide array of applications. We reformulated the second-stage problems using BDDs in order to apply Benders decomposition. We proposed a novel reformulation that parameterized the BDD arc costs using first-stage solutions and generalized an existing setting for the BDD reformulation from [29]. We analyzed the strength of these BDD-based Benders cuts and found they were incomparable but that both dominated the more traditional integer L-shaped cuts. We then extended the results to a risk-averse setting, presenting the first decomposition method for 2SP with binary recourse and CVaR. Finally, we proposed the SMWDS problem as an application for these methods and showed via computational experiments that our novel BDD-based algorithm performs best in practice.

We acknowledge that one of the main limitations of this BDD-based approach is the time spent to build the BDDs, thus it may be better suited to stochastic problems where a smaller number of scenarios are required. As a future research direction, we propose applying this methodology in a non-linear setting, such as the submodular optimization example seen in Section 1.4.6. There is also promise in applying these methods to problems where PB cuts are weak or cannot be generated, and to applications in the risk-averse and robust optimization settings.

References

[1] Shabbir Ahmed, Mohit Tawarmalani, and Nikolaos V Sahinidis. A finite branch-and-bound algorithm for two-stage stochastic integer programs. *Mathematical Programming*, 100(2):355–377, 2004.

[2] K. M. Alzoubi, Peng-Jun Wan, and O. Frieder. New distributed algorithm for connected dominating set in wireless ad hoc networks. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, pages 3849–3855, 2002.

[3] Gustavo Angulo, Shabbir Ahmed, and Santanu S. Dey. Improving the integer L-shaped method. *INFORMS Journal on Computing*, 28(3):483–499, 2016.

[4] Ayşe N. Arslan and Boris Detienne. Decomposition-based approaches for a class of two-stage robust binary optimization problems. *INFORMS Journal on Computing*, 2021.

[5] David Bergman and Andre A. Cire. Discrete nonlinear optimization by state-space decompositions. *Management Science*, 64(10):4700–4720, 2018.

[6] David Bergman, Andre A. Cire, Willem-Jan Van Hoeve, and John Hooker. *Decision Diagrams for Optimization*, volume 1. Springer, 2016.

[7] Nicolas Boria, Cécile Murat, and Vangelis Th. Paschos. The probabilistic minimum dominating set problem. *Discrete Applied Mathematics*, 234:93–113, 2018.

[8] Salim Bouamama and Christian Blum. A hybrid algorithmic model for the minimum weight dominating set problem. *Simulation Modelling Practice and Theory*, 64:57–68, 2016.

[9] Yann Bramoullé and Rachel Kranton. Public goods in networks. *Journal of Economic Theory*, 135(1):478–494, 2007.

[10] Claus C. Carøe and Rüdiger Schultz. Dual decomposition in stochastic integer programming. *Operations Research Letters*, 24(1-2):37–45, 1999.

[11] Claus C. Carøe and Jørgen Tind. L-shaped decomposition of two-stage stochastic programs with integer recourse. *Mathematical Programming*, 83(1):451–464, 1998.

[12] Margarita P. Castro, Andre A. Cire, and J. Christopher Beck. A combinatorial cut-and-lift procedure with an application to 0–1 second-order conic programming. *Mathematical Programming*, pages 1–57, 2021.

[13] Stefano Coniglio, Fabio Furini, and Ivana Ljubić. Submodular maximization of concave utility functions composed with a set-union operator with applications to maximal covering location problems. *Mathematical Programming*, 2022.

[14] Marc Demange, Marcel Adonis Haddad, and Cécile Murat. The probabilistic k-center problem. In *Proceedings of the Robust Solutions for Fire Fighting (RSFF) Workshop*, pages 62–74, 2018.

[15] Justin Dumouchelle, Rahul Patel, Elias B. Khalil, and Merve Bodur. Neur2SP: Neural two-stage stochastic programming. In *Proceedings of the 36th Annual Conference on Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[16] 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.

[17] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[18] Willem K. Klein Haneveld and Maarten H. van der Vlerk. Stochastic integer programming: General models and algorithms. *Annals of Operations Research*, 85:39, 1999.

[19] Utz-Uwe Haus, Carla Michini, and Marco Laumanns. Scenario aggregation using binary decision diagrams for stochastic programs with endogenous uncertainty. *arXiv preprint arXiv:1701.04055*, 2017.

[20] Jing (Selena) He, Zhipeng Cai, Shouling Ji, Raheem Beyah, and Yi Pan. A genetic algorithm for constructing a reliable mcds in probabilistic wireless networks. In Yu Cheng, Do Young Eun, Zhiguang Qin, Min Song, and Kai Xing, editors, *Wireless Algorithms, Systems, and Applications*, pages 96–107, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[21] J. N. Hooker and G. Ottosson. Logic-based Benders decomposition. *Mathematical Programming*, 96(1):33–60, 2003.

[22] John N. Hooker. Stochastic decision diagrams. In *CPAIOR Proceedings*, 2022.

[23] Raka Jovanovic, Milan Tuba, and Dana Simian. Ant colony optimization applied to minimum weight dominating set problem. In *Proceedings of the 12th WSEAS International Conference Control, Modeling and Simulation on Automatic Control, Modelling & Simulation*, pages 322–326, 2010.

[24] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 137–146, 2003.

[25] Simge Küçükyavuz and Suvrajeet Sen. An introduction to two-stage stochastic mixed-integer programming. In *Leading Developments from INFORMS Communities*, pages 1–27. INFORMS, 2017.

[26] 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.

[27] Can Li and Ignacio E. Grossmann. A generalized Benders decomposition-based branch and cut algorithm for two-stage stochastic programs with nonconvex constraints and mixed-binary first and second stage variables. *Journal of Global Optimization*, 75(2):247–272, 2019.

[28] Geng Lin, Jian Guan, and Huibin Feng. An ILP based memetic algorithm for finding minimum positive influence dominating sets in social networks. *Physica A: Statistical Mechanics and its Applications*, 500:199–209, 2018.

[29] Leonardo Lozano and J. Cole Smith. A binary decision diagram based algorithm for solving a class of binary two-stage stochastic programs. *Mathematical Programming*, 191(1):381–404, 2022.

[30] 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.

[31] Nilay Noyan. Risk-averse two-stage stochastic programming with an application to disaster management. *Computers & Operations Research*, 39(3):541–559, 2012.

[32] Lewis Ntaimo. Disjunctive decomposition for two-stage stochastic mixed-binary programs with random recourse. *Operations Research*, 58(1):229–243, 2010.

[33] Lewis Ntaimo. Fenchel decomposition for stochastic mixed-integer programming. *Journal of Global Optimization*, 55(1):141–163, 2013.

[34] Lewis Ntaimo and Suvrajeet Sen. A branch-and-cut algorithm for two-stage stochastic mixed-binary programs with continuous first-stage variables. *International Journal of Computational Science and Engineering*, 3(3):232–241, 2007.

[35] Lewis Ntaimo and Matthew W. Tanner. Computations with disjunctive cuts for two-stage stochastic mixed 0-1 integer programs. *Journal of Global Optimization*, 41(3):365–384, 2008.

[36] Yunwei Qi and Suvrajeet Sen. The ancestral Benders' cutting plane algorithm with multi-term disjunctions for mixed-integer recourse decisions in stochastic programming. *Mathematical Programming*, 161(1):193–235, 2017.

[37] R. Ravi and Amitabh Sinha. Hedging uncertainty: Approximation algorithms for stochastic optimization problems. *Mathematical Programming*, 108(1):97–114, 2006.

[38] R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–42, 2000.

[39] Andréa Santos, Fatiha Bendali, Jean Mailfert, Christophe Duhamel, and Kean-Mean Hou. Heuristics for designing energy-efficient wireless sensor network topologies. *Journal of Networks*, 2009.

[40] Rüdiger Schultz, Leen Stougie, and Maarten H. Van Der Vlerk. Two-stage stochastic integer programming: a survey. *Statistica Neerlandica*, 50(3):404–416, 1996.

[41] Rüdiger Schultz, Leen Stougie, and Maarten H. Van Der Vlerk. Solving stochastic programs with integer recourse by enumeration: A framework using Gröbner basis. *Mathematical Programming*, 83(1):229–252, 1998.

[42] Rüdiger Schultz and Stephan Tiedemann. Conditional value-at-risk in stochastic programs with mixed-integer recourse. *Mathematical Programming*, 105(2):365–386, 2006.

[43] Suvrajeet Sen. Algorithms for stochastic mixed-integer programming models. *Handbooks in Operations Research and Management Science*, 12:515–558, 2005.

[44] Suvrajeet Sen. Stochastic mixed-integer programming algorithms: Beyond Benders' decomposition. *Wiley Encyclopedia of Operations Research and Management Science*, 2010.

[45] Suvrajeet Sen and Julia L. Higle. The C^3 theorem and a D^2 algorithm for large scale stochastic mixed-integer programming: Set convexification. *Mathematical Programming*, 104(1):1–20, 2005.

[46] Suvrajeet Sen and Hanif D. Sherali. Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. *Mathematical Programming*, 106(2):203–223, 2006.

[47] Thiago Serra, Arvind U. Raghunathan, David Bergman, John Hooker, and Shingo Kobori. Last-mile scheduling under uncertainty. In Louis-Martin Rousseau and Kostas Stergiou, editors, *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 519–528. Springer International Publishing, 2019.

[48] Hanif D. Sherali and Barbara M. P. Fraticelli. A modification of Benders' decomposition algorithm for discrete subproblems: An approach for stochastic programs with integer recourse. *Journal of Global Optimization*, 22(1):319–342, 2002.

[49] Hanif D. Sherali and Xiaomei Zhu. On solving discrete two-stage stochastic programs having mixed-integer first- and second-stage variables. *Mathematical Programming*, 108(2):597–616, 2006.

[50] Ou Sun and Neng Fan. The probabilistic and reliable connected power dominating set problems. *Optimization Letters*, 13(5):1189–1206, 2019.

[51] Taro Takaguchi, Takehisa Hasegawa, and Yuichi Yoshida. Suppressing epidemics on networks by exploiting observer nodes. *Phys. Rev. E*, 90:012807, Jul 2014.

[52] Javad Akbari Torkestani and Mohammad Reza Meybodi. Finding minimum weight connected dominating set in stochastic graph based on learning automata. *Information Sciences*, 200:57–77, 2012.

[53] E. Ruben van Beesten and Ward Romeijnders. Convex approximations for two-stage mixed-integer mean-risk recourse models with conditional value-at-risk. *Mathematical Programming*, 181(2):473–507, 2020.

[54] Niels van der Laan and Ward Romeijnders. A converging Benders' decomposition algorithm for two-stage mixed-integer recourse models. *University of Groningen, SOM Research School*, 2020016-OPERA, 2020.

[55] Ingo Wegener. *Branching Programs and Binary Decision Diagrams*. Society for Industrial and Applied Mathematics, 2000.

[56] Jie Wu and Hailan Li. A dominating-set-based routing scheme in ad hoc wireless networks. *Telecommunication Systems*, 18(1):13–36, 2001.

[57] 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.