Solving Multi-Follower Mixed-Integer Bilevel Problems with Binary Linking Variables

Vladimir Stadnichuk[1], Arie Koster[2]

[1] School of Business and Economics, RWTH Aachen University

[2] Research Group on Discrete Optimization, RWTH Aachen University

vladimir.stadnichuk@om.rwth-aachen.de [1]
koster@math2.rwth-aachen.de [2]

December 29, 2024

Abstract

We study multi-follower bilevel optimization problems with binary linking variables where the second level consists of many independent integer-constrained subproblems. This problem class not only generalizes many classical interdiction problems but also arises naturally in many network design problems where the second-level subproblems involve complex routing decisions of the actors involved. We propose a novel branch-and-cut decomposition method that starts by solving the first level and then iteratively generates second-level feasibility and optimality cuts that are obtained by solving a slightly adjusted second-level problem. Compared to many other existing solution methods, we do not rely on solving the High Point Relaxation of the bilevel problem but fully project out the second level, resulting in significant computational advantages when the second-level problem is very large or possesses a weak LP relaxation. Also, our approach can be fully automated within a MIP solver, making it very easy to apply for those who do not want to design problem-tailored algorithms for their bilevel problem. Computational results for a bilevel network design problem demonstrate that our approach efficiently solves instances with hundreds of subproblems in a few minutes, significantly outperforming the Benders-like decomposition from the literature on challenging instances.

1 Introduction

Bilevel optimization is a highly popular tool to model hierarchical decision-making processes. It involves actors acting on two levels, referred to as the first and second level, acting within a hierarchical setting where each actor behaves according to its individual objective function. The first level makes the initial decisions, and the second level reacts by adjusting its strategy, which then influences the first-level pay-off. The objective is to find the first-level strategy that maximizes pay-off.

In this work, we focus on (optimistic) bilevel problems with binary linking variables. Within this setting, the linking variables are associated with resources, e.g., arcs within a network, shared by both levels, where the first level decides which of these resources are made available to the second level. This class of bilevel problems captures many existing optimization problems including (generalized) discrete interdiction games (Caprara et al., 2016; Della Croce & Scatamacchia, 2020; Fischetti et al., 2018; Israeli & Wood, 2002; Mattia, 2024), blocking problems (Bazgan et al., 2012; Mahdavi Pajouh, 2019), variants of the facility location and prize-collecting problems (cf. Fischetti et al., 2019), and various network design problems with routing decisions (Amaldi et al., 2011; Arslan et al., 2018; Cerulli et al., 2024; Fontaine & Minner, 2014, 2016; Gao et al., 2005; Kara & Verter, 2004; Marković et al., 2014, 2017).

If the second level is an LP, the bilevel optimization problem can be solved by a state-of-the-art MIP solver by applying the strong duality theorem to reformulate the bilevel problem into a single-level formulation. Because the resulting single-level MIP combines the first level with both the primal and dual formulations of the second level into a single model, the obtained MIP can be challenging to solve. Recently, Byeon and Van Hentenryck (2022) showed that the reformulated MIP can be efficiently addressed by Benders decomposition. The resulting Benders subproblem can be decomposed into a sequence of MIPs where one first solves the second level problem with a penalty term and afterwards the first level problem conditioned to the second level reaction. This approach significantly reduces the size of the individual MIPs while also preserving most of the structure of the first and second level in the intermediate solution steps, overall resulting in a highly effective, easy to understand and easy to implement approach.

If the second level involves integer variables, it generally becomes a non-convex optimization problem, and the strong duality reformulation can no longer be employed. While there exist some computational methods and general purpose solvers that can tackle the MIP-MIP bilevel structure (Avraamidou & Pistikopoulos, 2019; Fischetti et al., 2017; Kleniati & Adjiman, 2015; Poirion et al., 2020; Tahernejad et al., 2020; Taninmis & Sinnl, 2022; Tavaslıoğlu et al., 2019; Wang & Xu, 2017; Xu & Wang, 2014), they are designed for those with special structures, are hard to implement, or expect an MIP representation of the second level, making them hard to apply to instances where the second-level problem is hard to formulate as an MIP or possesses a weak LP relaxation.

An alternative approach to tackle interdiction and blocking games is the use of Benders-like cuts (cf. Israeli, 1999; Kleinert et al., 2021). These cuts can be directly implemented as a cutting plane procedure within modern MIP solvers, making them particularly popular among application-oriented users who wish to apply bilevel optimization in practice without designing tailored algorithms for their specific problems. One advantage of this Benders-like decomposition is that the separation problem involves solving the second-level problem, which is often well understood. Additionally, they can be effectively applied within a multi-follower bilevel setting, where the second level decomposes into multiple independent subproblems. In such cases, Benders-like cuts can be

generated independently for each subproblem, leading to an overall stronger formulation. Despite these benefits, Benders-like cuts are still primarily applied to specific problem classes (Caprara et al., 2016; Della Croce & Scatamacchia, 2020; Fischetti et al., 2019), e.g., satisfying the downwards monotonicity property, as they require a good understanding of the underlying problem structure to derive high-quality big M values needed to make them computationally viable.

Against this background, we extend the Benders-like decomposition approach by a smart application of classical Benders and Dantzig-Wolfe decomposition to obtain what we call a Hierarchical Decomposition, which combines the strengths of all the involved decompositions. The Dantzig-Wolfe step is used to project out the challenging integer variables from the second level to strengthen the MIP formulation, while the Benders step projects out the exponential number of variables, making our approach easy to implement within modern MIP solvers as a cutting plane procedure. Compared to the Benders-like cuts approach, our Hierarchical Decomposition preserves the original but also offers multiple additional advantages: First, the Benders step significantly reduces the size of the initial MIP formulation that must be solved, while the Dantzig-Wolfe decomposition leads to the generation of significantly stronger cutting planes in each step. Second, we offer high flexibility in the implementation. On the one hand, our approach can be implemented within a MIP solver without any understanding of the underlying problem structure. On the other hand, we can often transfer specialized solvers for the second-level optimization problem to solve the Dantzig-Wolfe induced pricing problems, and we present a detailed theoretical study of when this transformation is possible. Third, as we build directly on the Benders and Dantzig-Wolfe decompositions, we can leverage various existing techniques from these well-studied decomposition methods to further improve computational performance.

The Hierarchical Decomposition can efficiently address multi-follower bilevel problems, for which algorithmic solutions are still sparse (cf. Tavaslıoğlu et al., 2019), by integrating it into a multi-cut Benders framework. To demonstrate these computational strengths, we apply the Hierarchical Decomposition to bilevel network design problems involving a large number of second-level actors making (complex) routing decisions. We provide an efficient implementation for the Hazmat Network Design Problem with Capacity Constraints (HNDPwCC), which generalizes many network design problems with resource constraints. Computational results show that our approach can solve instances with a few hundred second-level actors within a few minutes, while the original Benders-like decomposition requires more that ten times the runtime for challenging instances.

The remainder of this paper is structured as follows: In Section 2, we formally introduce the bilevel problem with binary linking variables and briefly recap the Benders-like cuts reformulation. The Hierarchical Decomposition is presented in Section 3. We discuss the technical details, including various speed-up techniques and structural analysis of the involved MIPs, in Section 4. The efficient implementation for the HNDPwCC is detailed in Section 5, where we also demonstrate how our approach can be efficiently extended to multi-follower bilevel problems. Computational

results are presented in Section 6, and Section 7 offers concluding remarks.

2 Problem Setting

2.1 Bilevel Problems with Binary Linking Variables

We consider the following optimistic bilevel setting. Let $\mathcal{A}$ be the set of shared resources, and let x_a be the binary linking variables indicating whether resource a is available ($x_a = 1$) or not ($x_a = 0$) at the second level. We use bold notation to indicate vectors/matrices of variables/parameters, e.g., $\mathbf{x}$ represents the vector of all x-variables. The second level contains variables $0 \leq y_a \leq C_a$, $a \in \mathcal{A}$, that correspond to the consumption of resource a, and both levels are linked by the condition that $x_a = 0$ implies $y_a = 0$. Further, we allow for additional purely first level variables within $\mathcal{H}(\mathbf{x}) = \{\mathbf{h} : \mathbf{W_h h} + \mathbf{W_x x} \geq \mathbf{w}; \mathbf{h} \in \mathbb{R}^{n_h} \times \mathbb{N}^{m_h}\}$ and purely second level variables within $\mathcal{Z}(\mathbf{y}) = \{\mathbf{z} : \mathbf{V_y y} + \mathbf{V_z z} \geq \mathbf{v}; \mathbf{z} \in \mathbb{R}^{n_z} \times \mathbb{N}^{m_z}\}$. Especially, the second level variables $\mathbf{z} \in \mathcal{Z}(\mathbf{y})$ can be integer.

Assuming that the second level evaluates its decisions based on a (linear) objective function $c(\mathbf{y}, \mathbf{z})$, and the first level evaluates the pay-off based on (linear) functions $r(\mathbf{h}, \mathbf{x})$ for their own decisions and $r(\mathbf{y}, \mathbf{z})$ for the second-level decisions, we can formally state our bilevel setting as

$$\min \quad r(\mathbf{h}, \mathbf{x}) + r(\mathbf{y}, \mathbf{z}) \tag{1a}$$
$$\mathbf{h} \in \mathcal{H}(\mathbf{x}) \tag{1b}$$
$$x_a \in \{0, 1\} \qquad \forall a \in \mathcal{A} \tag{1c}$$
$$(\mathbf{y}, \mathbf{z}) \in \mathcal{S}(\mathbf{x}) \tag{1d}$$

where $\mathcal{S}(\mathbf{x})$ is the set of optimal solutions for a fixed $\mathbf{x}$ of the second level

$$\min \quad c(\mathbf{y}, \mathbf{z}) \tag{2a}$$
$$\text{s.t.} \quad y_a \leq C_a x_a \qquad \forall a \in \mathcal{A} \tag{2b}$$
$$y_a \geq 0 \qquad \forall a \in \mathcal{A} \tag{2c}$$
$$\mathbf{z} \in \mathcal{Z}(\mathbf{y}) \tag{2d}$$

Remark 1. The above partition of the second level into two variables, $\mathbf{y}$ and $\mathbf{z}$, is only for our convenience, as it allows us to present the following ideas more clearly. As we discuss in Appendix A, we can transform any general second-level problem into the above interdiction structure with the $\mathbf{y}$-variables. Specifically, the $\mathbf{y}$ can also be subject to some integrality constraints, e.g., by adding constraints $\mathbf{y} = \mathbf{z}$ to the second level and forcing $\mathbf{z}$ to be integer.

Remark 2. Note that the above bilevel formulation does not include coupling constraints, which are first-level constraints that explicitly depend on second-level variables. While we can handle coupling constraints to some extent, they introduce complex notation and case distinctions. There-

fore, we present our main ideas using the formulation without coupling constraints but discuss their impact in Appendix B.

Also, we make the following assumption within the remainder of this paper:

Assumption 1. Both $\mathcal{H}(\mathbf{x})$ and $\mathcal{Z}(\mathbf{y})$ are bounded, and C_a are finite for all $a \in \mathcal{A}$. Therefore, the first level is bounded and the second level is either bounded or infeasible, depending on the selected first level solution.

Kleinert et al. (2021) and Xu and Wang (2014) provide an in detail discussion on why this is necessary for the Benders-like cuts approach that we discuss next.

2.2 Recap: Benders-like Cuts

Note that MIP-MIP bilevel problems are typically Σ_2^P-hard (Jeroslow, 1985). This means the problem can be solved in nondeterministic polynomial time given an oracle that solves the second level in constant time. Therefore, we cannot expect to find a compact single-level formulation if $\mathbf{y}$ or $\mathbf{z}$ variables have integrality restrictions. Instead, we can first solve the High Point Relaxation (HPR), i.e., the problem variant where the second-level objective is ignored and full cooperation between the two levels is assumed. We then sequentially add a potentially exponential number of Benders-like cuts to ensure that the found solution is optimal with respect to the second-level objective function.

To obtain the Benders-like cuts, assume that for a given $\mathbf{x}$, the second level contains a feasible solution. Then we can find coefficients M_a such that Model (2) is equivalent to

$$\min \quad c(\mathbf{y}, \mathbf{z}) + \sum_{a \in \mathcal{A}} M_a (1 - x_a) y_a \tag{3a}$$

$$\text{s.t.} \quad y_a \leq C_a \qquad \forall a \in \mathcal{A} \tag{3b}$$

$$y \geq 0 \qquad \forall a \in \mathcal{A} \tag{3c}$$

$$\mathbf{z} \in \mathcal{Z}(\mathbf{y}) \tag{3d}$$

The linking variables are shifted from the constraints (2b) to the objective function. If $x_a = 0$, the big M cost M_a are set so high that any feasible solution with $y_a > 0$ is no longer optimal.

Remark 3. Computing good big M values is highly challenging even when the second level is an LP (Kleinert & Schmidt, 2023; Kleinert et al., 2020). This is also why applying Benders-like cuts to general bilevel problems with binary linking constraints is hard to automate, as using a trial-and-error approach to find these large constants can lead to highly suboptimal solutions (Pineda & Morales, 2019).

As the feasibility region $\mathcal{P} = conv\{(\mathbf{y}, \mathbf{z}); y_a \leq C_a \, \forall a \in \mathcal{A}; \mathbf{y} \geq 0; \mathbf{z} \in \mathcal{Z}(\mathbf{y})\}$ of Model (3) is again a polytope that no longer depends on the x-variables, we can enumerate all of its extreme

points $\overline{\mathcal{P}}$. The notation *conv* indicates that when $\mathcal{H}(\mathbf{x})$ or $\mathcal{Z}(\mathbf{y})$ restrict variables to be integer, $\overline{\mathcal{P}}$ also contains only the necessary integer points. To link the extreme points $p \in \overline{\mathcal{P}}$ with the original y_a and $\mathbf{z}$ variables, we write y_a^p as the components of y_a, and $\mathbf{z}^p$ as the components of z, in p. Also, we define $c_p := c(\mathbf{y}^p, \mathbf{z}^p)$ as the cost of point p with respect to the second-level objective function. With this notation, we can combine Model (1) with the constraints from Model (2) into the following Benders-like reformulation

$$\begin{aligned}
\min \quad & r(\mathbf{h}, \mathbf{x}) + r(\mathbf{y}, \mathbf{z}) && && \text{(4a)}\\
\text{s.t.} \quad & y_a \leq C_a x_a && \forall a \in \mathcal{A} && \text{(4b)}\\
& c(\mathbf{y}, \mathbf{z}) \leq c_p + \sum_{a \in \mathcal{A}} M_a (1 - x_a) y_a^p && \forall p \in \overline{\mathcal{P}} && \text{(4c)}\\
& \mathbf{h} \in \mathcal{H}(\mathbf{x}) && && \text{(4d)}\\
& x_a \in \{0, 1\} && \forall a \in \mathcal{A} && \text{(4e)}\\
& \mathbf{y} \geq 0 && \forall a \in \mathcal{A} && \text{(4f)}\\
& \mathbf{z} \in \mathcal{Z}(\mathbf{y}) && && \text{(4g)}
\end{aligned}$$

Constraints (4b) and (4d)-(4g) ensure that we select a solution $(\mathbf{h}, \mathbf{x}, \mathbf{y}, \mathbf{z})$ that is both first- and second-level feasible. Note that constraints (4b) are the original second-level constraints (2b) where the $\mathbf{x}$-variables explicitly forbid the $\mathbf{y}$-variables. The Benders-like cuts (4c) then enforce that the selected solution is also optimal with respect to the second-level objective.

Model (4) combines an easy-to-implement framework with the advantage that generating the Benders-like cuts (4c) requires solving the second level, for which fast, specialized algorithms are often available (cf. Fischetti et al., 2019; Furini et al., 2019). However, the initial HPR, i.e., Model (4) without Benders-like cuts (4c), suffers from a poor LP relaxation (Moore & Bard, 1990).

One often ignored, but important additional challenge, is that we must explicitly include the second-level constraints (4g) in the Benders-like cuts reformulation to ensure that the found first-level solutions are feasible for the second level. These additional constraints are undesirable for two reasons. First, we often want to apply specialized algorithms to solve the second-level problem, while the MIP formulation with constraints (4g) may suffer from a poor LP relaxation. This leads to the second problem: we have to invest a significant amount of additional computational effort to solve the HPR, e.g., by branching on the y or z variables or generating cutting planes to strengthen the current formulation, in order to find some second-level feasible solutions. Such solutions are often directly cut off by the Benders-like cuts (4c), as they are not second-level optimal.

3 Hierarchical Decomposition

To address the mentioned drawbacks, we extend the Benders-like cuts approach with additional decomposition methods from the literature. The necessary steps are shown in Figure 1. First, we employ Dantzig-Wolfe decomposition to project out the integer variables from the second level. As the resulting Dantzig-Wolfe reformulated model suffers from an exponential number of variables (and constraints), we use Benders decomposition to partition the first and second level parts. The Benders master problem preserves the first-level decisions, while the second-level variables are projected out into the Benders subproblem. To deal with the exponential number of variables in the Benders subproblem, we find a polynomial subset of these variables that provably yield an optimal solution. The resulting reduced Benders subproblem still contains an exponential number of constraints that can be separated by an auxiliary version of the second-level problem where the use of certain resources is penalized.

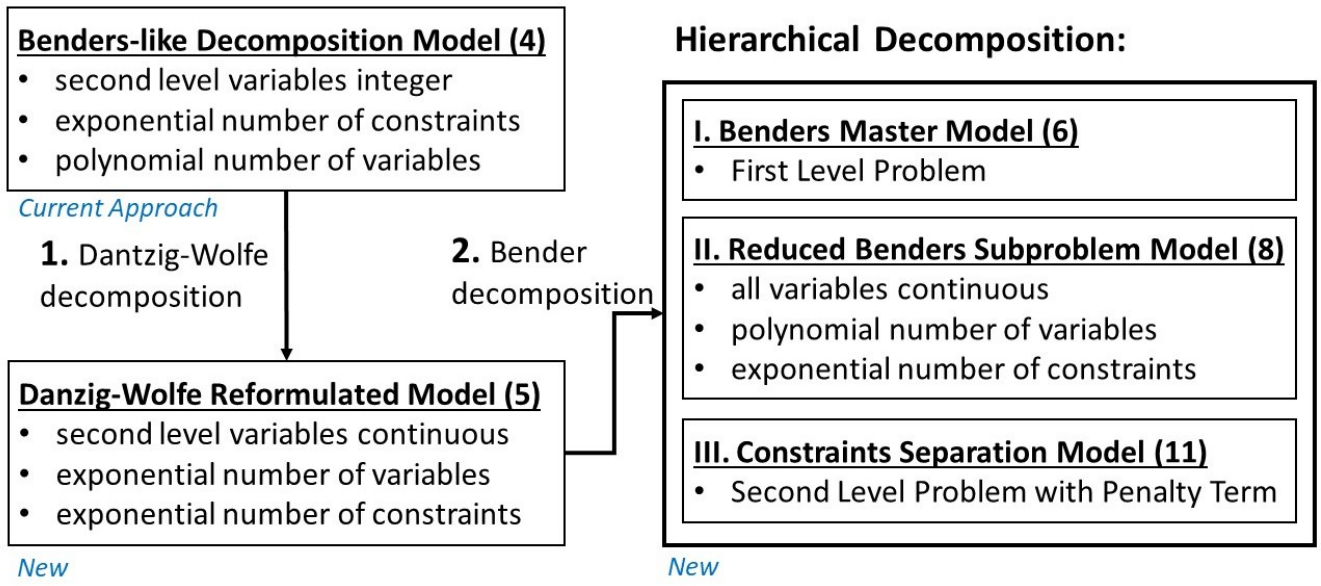

Figure 1: Decomposition steps employed to obtain the Hierarchical Decomposition.

As our new approach fully decomposes the hierarchical structure within the two levels into independent subproblems, we refer to it as *Hierarchical Decomposition*. This new approach extends the results from Byeon and Van Hentenryck (2022) with convex followers to our setting where the second level can contain integer variables.

From a computational point of view, this new approach offers multiple benefits:

- By fully projecting out the second level from the Benders master problem, we significantly reduce its size and computational complexity compared to the previously necessary HPR. This is especially advantageous if the second-level problem is hard to formulate as an MIP

or possesses a weak LP relaxation, while it can be efficiently solved with, e.g., dynamic or constraint programming.

- The constraint separation model strongly preserves the structure of the second level. Therefore, theoretical and algorithmic results can be directly transferred from the second level—which is often a well-understood optimization problem—to the constraint separation model.
- The generated Benders optimality and feasibility cuts are far less dependent on the involved big M constants, and we show later how numeric stable bounds for the cut coefficients can be obtained with (low) computational effort.
- Because our Hierarchical Decomposition is based directly on classical Benders decomposition, we are able to transfer many well-known speed-up techniques like partial decomposition or Pareto-optimal Benders cuts that further improve runtime.
- If the second level decomposes into multiple independent subproblems, i.e., we have a multi-follower bilevel problem, we can extend Step 2 to a multi-cut Benders decomposition to generate optimality and feasibility cuts independently for each second-level actor.

Next, we present the technical details for each decomposition step and point out structural properties that enable a more efficient implementation.

4 Danzig-Wolfe-like Decomposition

In this section, we detail each step of the Hierarchical Decomposition. Step 1 including the Dantzig-Wolfe reformulated model is presented in Section 4.1, while the Benders decomposition (Step 2) is shown in Section 4.2. Section 4.3 focuses on the constraint separation model, and in Section 4.4 we briefly discuss how to address the big M contained in the generated cut coefficients.

4.1 Dantzig-Wolfe Decomposition

We apply a Dantzig-Wolfe decomposition (Desrosiers & Lübbecke, 2006) by aggregating the y and z variables into the extreme point variables $f_p \in \{0,1\}$ for $p \in \overline{\mathcal{P}}$ (see Section 2.2). We naturally extend the notation $r_p = r(\mathbf{y}^p, \mathbf{z}^p)$ to represent the cost component of p in the first-level objective. We also write $\overline{\mathcal{P}}_a = \{p \in \overline{\mathcal{P}} : y_a^p > 0\}$ for a fixed $a \in \mathcal{A}$ as the subset of all extreme points where resource a is used, and $\mathcal{A}_p = \{a \in \mathcal{A} : y_a^p > 0\}$ for the set of resources that are used in solution $p \in \overline{\mathcal{P}}$. Note that $p \in \overline{\mathcal{P}}_a$ if and only if $a \in \mathcal{A}_p$. Applying the Dantzig-Wolfe decomposition step on Model (4), we obtain

$$\min \quad r(\mathbf{h},\mathbf{x}) + \sum_{p\in\overline{\mathcal{P}}} r_p f_p \tag{5a}$$

$$\text{s.t.} \quad \sum_{p\in\overline{\mathcal{P}}} f_p = 1 \tag{5b}$$

$$\sum_{p\in\overline{\mathcal{P}}_a} f_p \le C_a x_a \qquad \forall a \in \mathcal{A} \tag{5c}$$

$$\sum_{p'\in\overline{\mathcal{P}}} c_{p'} f_{p'} \le c_p + \sum_{a\in\mathcal{A}} M_a(1-x_a)y_a^p \qquad \forall p \in \overline{\mathcal{P}} \tag{5d}$$

$$f_p \ge 0 \qquad \forall p \in \overline{\mathcal{P}} \tag{5e}$$

$$\mathbf{h} \in \mathcal{H}(\mathbf{x}) \tag{5f}$$

$$x_a \in \{0,1\} \qquad \forall a \in \mathcal{A} \tag{5g}$$

Constraints (5b)-(5d) ensure that the extreme point $p \in \overline{\mathcal{P}}$ that is second-level optimal for the chosen x-variables is selected. Because these constraints model the simple selection of the f_p-variable where $p \in \overline{\mathcal{P}}$ is feasible for the current x-variable selection and has the lowest cost among them, we do not need to enforce integrality on the f-variables.

Because $\overline{\mathcal{P}}$ are the extreme points of the convex hull of the second-level problem, we expect the LP relaxation of Model (5) to be stronger than that of the original Benders-like cuts Model (4). However, this additional strength comes at the expense of an exponential number of both variables and constraints. For a solution approach that does not require an exponential number of variables, observe that Model (5) can be partitioned into first-level constraints (5f) and (5g), which include integer variables, and second-level constraints (5b)-(5d), which induce an LP when parameterized by the $\mathbf{x}$-variables. This structure motivates the application of (classical) Benders decomposition.

4.2 Benders Decomposition

The Benders decomposition projects out the inner optimization problem $\Phi(x) = \min_{\mathbf{f}} \left\{ \sum_{p\in\overline{\mathcal{P}}} r_p f_p \mid (5\text{b}) - (5\text{d}); f_p \ge 0 \, \forall p \in \overline{\mathcal{P}} \right\}$, where $\Phi(x)$ is the optimal objective value for given x. By Assumption 1, there exists an L with $\Phi(x) \ge L$. Model (5) can then be rewritten into

$$\min \quad r(\mathbf{h},\mathbf{x}) + \sigma \tag{6a}$$

$$\text{s.t.} \quad \mathbf{h} \in \mathcal{H}(\mathbf{x}) \tag{6b}$$

$$x_a \in \{0,1\} \qquad \forall a \in \mathcal{A} \tag{6c}$$

$$\sigma \ge \Phi(x) \tag{6d}$$

$$\sigma \ge L \tag{6e}$$

Here, the auxiliary variable σ replaces $\sum_{p\in\overline{\mathcal{P}}} r_p f_p$ in the objective function. Because the inner optimization problem is an LP, we can replace $\Phi(x)$ by a set of linear constraints obtained from the dual problem. Let s, k_a, and g_p be the dual variables of constraints (5b), (5c), and (5d), respectively. To simplify notation, we also define $\theta_p(x) = c_p + \sum_{a\in\mathcal{A}} M_a(1-x_a)y_a^p$. For a fixed $\overline{\mathbf{x}}$, the dual of the inner optimization problem is then

$$\max \quad s - \sum_{a\in\mathcal{A}} C_a\overline{x}_a k_a - \sum_{p\in\overline{\mathcal{P}}} \theta_p(\overline{x}) g_p \tag{7a}$$

$$\text{s.t.} \quad s - \sum_{a\in\mathcal{A}_p} k_a - c_p \sum_{p'\in\overline{\mathcal{P}}} g_{p'} \le r_p \qquad \forall p\in\overline{\mathcal{P}} \tag{7b}$$

$$s \text{ free} \tag{7c}$$

$$k_a \ge 0 \qquad \forall a\in\mathcal{A} \tag{7d}$$

$$g_p \ge 0 \qquad \forall p\in\overline{\mathcal{P}} \tag{7e}$$

Because of Assumption 1, the feasibility region of Model (7) is non-empty and does not depend on $\overline{\mathbf{x}}$. Therefore, it can be partitioned into a set of extreme rays $\mathcal{C}$ and extreme points $\mathcal{O}$. The Benders decomposition approach then iteratively by first solving the relaxed Model (6) without constraints (6d), which is called the Benders master problem in the literature. Model (7) is then solved for the given $\overline{\mathbf{x}}$. If Model (7) is unbounded, it proofs that the original inner optimization problem was infeasible, and we add an constraint that limits the movement in the direction of the associated extreme ray. Otherwise, we compute the optimal solution $\Phi(\overline{\mathbf{x}})$, and add a constraint ensuring that $\sigma \ge \Phi(\overline{\mathbf{x}})$ if $\overline{\mathbf{x}}$ is selected as solution.

The structure of the generated constraints depends on that of the extreme rays $\mathcal{C}$ and extreme points $\mathcal{O}$. While Model (7) posses an exponential number of variables, we are able to show that most of these variables are zero within an optimal solution (or unbounded ray).

Lemma 1 (Reduced Benders Subproblem)**.** Let $\overline{\mathbf{x}}$ be fix and let Model (7) be bounded for this $\overline{\mathbf{x}}$. Then $\mathcal{S}(\overline{\mathbf{x}})$ is non-empty, and for all $p^* \in \mathcal{S}(\overline{\mathbf{x}}) \cap \overline{\mathcal{P}}$ there exists an optimal solution $(s, \mathbf{k}, \mathbf{g}) \in \mathcal{O}$ to Model (7) with $g_p = 0$ for all $p \ne p^*$.

Proof. Proof First note that because of Assumption 1 and the fact that Model (7) is bounded, there exists a feasible second level solution for $\overline{\mathbf{x}}$, i.e., $\mathcal{S}(\overline{\mathbf{x}})$ is non empty. Fix an arbitrary $p^* \in \mathcal{S}(\overline{\mathbf{x}})$. By definition of $\mathcal{S}(\overline{\mathbf{x}})$, $f_{p^*} = 1$ and $f_p = 0; \forall p \ne p^*$ is an optimal solution for the inner optimization problem $\min_{\mathbf{f}} \left\{ \sum_{p\in\overline{\mathcal{P}}} r_p f_p \mid (5b) - (5d); f_p \ge 0 \, \forall p \in \overline{\mathcal{P}} \right\}$, where the first level decisions are fixed to $\overline{x}$. For this solution, the Benders-like cut $\sum_{p'\in\overline{\mathcal{P}}} c_{p'} f_{p'} \le c_{p^*} + \sum_{a\in\mathcal{A}} M_a(1-x_a)y_a^{p^*}$ corresponding to p^* is tight, i.e., satisfied by equality. All other Benders-like cuts are redundant for the optimality in the sense that they are either not tight for this solution, or correspond to $p \in \overline{\mathcal{P}}$ with the same second

level objective value as p^*. By a complementary slackness argument, the dual variables g_p with $p \in \overline{\mathcal{P}}$, $p \neq p^*$ are also redundant and can be assumed to be 0. □ □

Lemma 2 (Extreme Ray Classification)**.** Let $\overline{\mathbf{x}}$ be fix and let Model (7) be unbounded. Then we can find an unbounded ray $(s, \mathbf{k}, \mathbf{g}) \in \mathcal{C}$ with $s = 1$ and $\mathbf{g} = 0$.

Proof. Proof By duality theory, the inner optimization problem $\min_{\mathbf{f}} \left\{ \sum_{p \in \overline{\mathcal{P}}} r_p f_p \mid (5b) - (5d); f_p \geq 0 \, \forall p \in \overline{\mathcal{P}} \right\}$ is infeasible, as Model (7) is unbounded. To proof that $\mathcal{S}(\overline{\mathbf{x}})$ is empty, we only have to check weather there exists an $p \in \overline{\mathcal{P}}$ satisfying constraints (5b) and (5c). The minimal such p (with respect to the second level objective function) then also satisfies the Benders-like cuts constraints (5c). Therefore, constraints (5b) are irrelevant for the question of feasibility. By a complementary slackness argument, if Model (7) is unbounded, we can always find an unbounded ray $(s, \mathbf{k}, \mathbf{g}) \in \mathcal{C}$ with $\mathbf{g} = 0$. The lemma follows by rescaling this ray by $\frac{1}{s}$. □ □

Combining the results from Lemma 1 and 2, we first solve the second level (Model (2)) for a given master solution $\overline{\mathbf{x}}$, followed by solving the following reduced dual problem

$$\max \quad s - \sum_{a \in \mathcal{A}} C_a \overline{x}_a k_a - \theta_{p^*}(\overline{x}) g_{p^*} \tag{8a}$$

$$\text{s.t.} \quad s - \sum_{a \in \mathcal{A}_p} k_a - c^u_p g_{p^*} \leq r_p \qquad \forall p \in \overline{\mathcal{P}} \tag{8b}$$

$$s \text{ free} \tag{8c}$$

$$k_a \geq 0 \qquad \forall a \in \mathcal{A} \tag{8d}$$

$$g^u_{p^*} \geq 0 \tag{8e}$$

where p^* is the found optimal second level solution, or $g_{p^*} = 0$ if the second level is infeasible.

If Model (7) is unbounded, we find the extreme ray $(\overline{s}, \overline{\mathbf{k}}, \overline{\mathbf{g}}) \in \mathcal{C}$ with $\overline{s} = 1$, $\overline{\mathbf{k}} \geq 0$, and $\overline{\mathbf{g}} = 0$, resulting in a Benders feasibility cut

$$\sum_{a \in \mathcal{A}} \overline{k}_a x_a \geq 1 \tag{9}$$

that is added to Model (6) to cut of the current infeasible $\overline{\mathbf{x}}$ solution. These feasibility cuts closely resemble combinatorial Benders cuts (cf. Codato & Fischetti, 2006; Rahmaniani et al., 2017). The $\overline{x}_a$ with $\overline{k}_a > 0$ form a (minimal) set of interdicted variables that result in the infeasibility of the second level, and constraints (9) exclude such sets from the master problem.

If Model (5) is feasible, we find an optimal solution $(\overline{s}, \overline{\mathbf{k}}, \overline{g}_{p^*}) \in \mathcal{O}$ and add (if violated) an optimality constraint

$$\overline{s} - \theta_{p^*}(\overline{x}) \overline{g}_{p^*} - \sum_{a \in \mathcal{A}} \overline{k}_a x_a \leq \sigma \tag{10}$$

that ensures that $\sigma \geq \sum_{p \in \overline{\mathcal{P}}} r_p f_p$ holds. Replacing $\theta_p(x)$ by its definition results in the cutting plane to be added.

The advantage of the Benders approach compared to Model (5) is that we only need an exponential number of constraints, i.e., we require only row generation and no column generation, which is significantly easier to implement in most current MIP solvers like Gurobi or IBM CPLEX. However, we do not fully remove the additional complexity of column generation but rather shift it from the Benders master to the subproblem (Model 7). Next, we discuss how we can still solve the Benders subproblem efficiently.

4.3 Constraints Separation Model

The exponential number of constraints (7b) can be addressed by an iterative separation procedure. Let $\overline{s}$, $\overline{k}_a$, and $\overline{g}_{p^*}$, be the current Model (8) solution. If Model (8) is unbounded, let it be the undounded ray with $\overline{g}_{p^*} = 0$. Then, we find a violated constraint (7b) by solving the following auxiliary MIP:

$$\min \quad c(\mathbf{y}, \mathbf{z})\overline{g}_{p^*} + r(\mathbf{y}, \mathbf{z}) + \sum_{a \in \mathcal{A}} \overline{k}_a \phi_a \tag{11a}$$

$$\text{s.t.} \quad y_a \leq C_a \phi_a \qquad \forall a \in \mathcal{A} \tag{11b}$$

$$\phi_a \in \{0, 1\} \qquad \forall a \in \mathcal{A} \tag{11c}$$

$$\mathbf{y} \geq 0 \tag{11d}$$

$$\mathbf{z} \in \mathcal{Z}(\mathbf{y}) \tag{11e}$$

If the found solution $(\overline{y}, \overline{z}, \overline{\phi})$, with $\overline{p}$ being the extreme point associated with $(\overline{y}, \overline{z})$, has value smaller than $\overline{s}$, the constraint $\overline{s} - \sum_{a \in \mathcal{A}_{\overline{p}}} \overline{k}_a - c_{\overline{p}}\overline{g}_{p^*} \leq r_{\overline{p}}$ is violated. Otherwise, we have proven optimality of the solution.

If we find a violated constraint (7b), we add it to Model (8), resolve its, and then run Model (11) with the new obtained solution again. To ensure that we do not generate the constraints associated with the same extreme point $p \in \overline{\mathcal{P}}$ twice, we employ the following warm start procedure.

Remark 4 (Warm Start)**.** Observe that only the objective function, but not the feasibility region, of Model (8) depends on the current master solution $\overline{x}$. Therefore, instead of constructing a new Benders subproblem for each new $\overline{x}$, we only change the objective function while retaining the constraints generated from previous runs.

Model (11) corresponds to the second-level Model (2) with an adjusted objective function where the coefficient vector $\mathbf{c}$ is replaced with $\mathbf{c} + \mathbf{r}$, and where resources $a \in \mathcal{A}$ are not explicitly allowed or forbidden, but costs k_a are charged for usage. This changes the problem from a classical

interdiction setting to a pricing problem, where Model (8) sets the resource prices and then Model (11) decides whether to buy these resources or not.

Assuming that we have an efficient algorithm for solving the second level problem, we ask the question whether Model (11) can be solved with the same algorithm, and if not, how the complexity class of the underlying problem changes. We motivate this study with the following two results.

Lemma 3 (Elementary Shortest Path)**.** Let the second level be an Elementary Shortest Path Problem that asks for the shortest cycle-free path in the network. Then the Model (11) is again an Elementary Shortest Path Problem.

Proof. Proof Let consider that the second level level is a Elementary Shortest Path Problem on a network $(\mathcal{N}, \mathcal{A})$ where variables y_a indicate whether arc $a \in \mathcal{A}$ is contained in the shortest path or not. Let $o \in \mathcal{N}$ be the origin, $t \in \mathcal{N}$ the destination, and $\mathcal{S} = \{S \subseteq \mathcal{N} \backslash \{t\} : |S| \geq 2\}$ the set of node subsets necessary for the subtour elimination constraints. Then the Model (11) can be formulated as

$$\min \quad \sum_{a \in \mathcal{A}} (c_a + r_a) y_a + \sum_{a \in \mathcal{A}} \overline{k}_a \phi_a \tag{12a}$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(i)} y_a - \sum_{a \in \delta^-(i)} y_a = d_i \qquad \forall i \in \mathcal{N} \tag{12b}$$

$$\sum_{a \in \delta^+(S)} y_a \geq \sum_{a \in \delta^+(i)} y_a \qquad \forall i \in S, S \in \mathcal{S} \tag{12c}$$

$$y_a \leq \phi_a \qquad \forall a \in \mathcal{A} \tag{12d}$$

$$\phi_a \in \{0, 1\} \qquad \forall a \in \mathcal{A} \tag{12e}$$

$$y_a \geq 0 \qquad \forall a \in \mathcal{A} \tag{12f}$$

where $d_o = 1$, $d_d = -1$, and otherwise $d_i = 0$ for all $i \in \mathcal{N} \backslash \{s, t\}$. Note that the subtour elimination constraints (12c) are only required when the graph with arc cost $(c_a + r_a) y_a + \overline{k}_a \phi_a$ contains negative cycles. As we only "buy" an arc a if we also use it in the shortest path, it holds $y_a = \phi_a$ for all $a \in \mathcal{A}$. Hence, the Model (11) reduces back to an elementary shortest path problem. □ □

The complexity of the Elementary Shortest Path Problem strongly depends on the existence of negative cycles. Depending on the initial cost structure of arc costs c_a and the new cost function $c_a + r_a + \overline{k}_a$, the complexity can jump between being polynomially solvable when no negative cycles exist and strongly NP-hard otherwise (cf. Boland et al., 2006).

We consider this a positive result, as the structure of the second-level problem directly transfers to that of Model (11). However, this is not always necessarily the case, as shown in the following.

Lemma 4 (Multi-Commodity Flow)**.** Let the second level be the Fractional Multi-Commodity Flow Problem. Then the Model (11) is the Multicommodity Capacitated Fixed-Charge Network Design.

Proof. Proof If the second level is the Fractional Multi-Commodity Flow Problem with $\mathcal{U}$ being the set of commodities, $(\mathcal{N}, \mathcal{A})$ the underlying network, z_a^u the continuous variable modeling the flow over arc a for commodity u, and y_u the auxiliary variable modeling the aggregated flow over arc a. We also make the common assumption that the arc-cost $c_a^u \geq 0$ are non-negative. Then, we can formulate Model (11) as follows:

$$\min \quad \sum_{u \in \mathcal{U}} \sum_{a \in \mathcal{A}} c_a^u z_a^u + \sum_{a \in \mathcal{A}} \overline{k}_a \phi_a \tag{13a}$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(i)} z_a^u - \sum_{a \in \delta^-(i)} z_a^u = d_i^u \qquad \forall u \in \mathcal{U}, i \in \mathcal{N} \tag{13b}$$

$$\sum_{u \in \mathcal{U}} z_a^u = y_a \qquad \forall a \in \mathcal{A} \tag{13c}$$

$$y_a \leq C_a \phi_a \qquad \forall a \in \mathcal{A} \tag{13d}$$

$$\phi_a \in \{0, 1\} \qquad \forall a \in \mathcal{A} \tag{13e}$$

$$y_a^u \geq 0 \qquad \forall u \in \mathcal{U}, a \in \mathcal{A} \tag{13f}$$

$$z_a^u \geq 0 \qquad \forall u \in \mathcal{U}, a \in \mathcal{A} \tag{13g}$$

Here, d_i^u are again the demand variables for each commodity $u \in \mathcal{U}$. This MIP model is exactly the Multicommodity Capacitated Fixed-Charge Network Design (Chouman et al., 2017). □ □

The Multicommodity Capacitated Fixed-Charge Network Design is known to be NP-hard, while the Fractional Multi-Commodity Flow Problem is polynomial-time solvable. This strong jump in the complexity of the problem can be traced back to the y_a-variables that link the resource cost with the remainder. In the Elementary Shortest Path Problem, these variables are binary (due to the total unimodular structure of the second level constraint matrix) and can therefore directly substitute the ϕ_a-variables. Conversely, in the Fractional Multi-Commodity Flow Problem, these variables link the binary investment decisions with the flow decisions of multiple commodities. In this latter case, it is the binary decisions ϕ_a that lead to a significant increase in complexity.

We generalize these observations in the following theorem.

Theorem 1 (Binary Second Level)**.** Let the second-level problem be given in the interdiction form presented in Model (2). If the second level variables $\mathbf{y}$ model only binary decisions, the Model (11) reduces to exactly the second level problem with all resources $a \in \mathcal{A}$ available and the objective function

$$c(\mathbf{y}, \mathbf{z}) \overline{g}_{p^*} + r(\mathbf{y}, \mathbf{z}) + \sum_{a \in \mathcal{A}} \overline{k}_a y_a$$

Proof. Proof If the $\mathbf{y}$ variables model only binary decisions, we conclude from constraints (11b) that $\phi_a = 1$ if and only if $y_a = 1$, as w.l.o.g. we would never invest in buying a if we do not require it in a solution. Therefore, we can replace any recurrence of ϕ_a with y_a, from which the theorem follows. □ □

Linking our results with the Dantzig-Wolfe decomposition, Theorem 1 classifies when constraints (5c) and (5d) are robust (cf. Clautiaux & Ljubić, 2024), i.e., preserve the difficulty of the pricing problem. Hence, results from the literature on the robustness of specific constraints can be transferred to our problem setting, and vice versa.

4.4 Improving big M Coefficients

As mentioned in Remark 3, it is computationally challenging to determine good values for the big M's contained in the optimality cuts (10). The main advantage of our approach is that the big M are required only for the theoretical reformulation steps, while alternative numerically stable coefficients for the cuts can be computed easily (in comparison).

Consider the auxiliary model

$$\begin{aligned}
\min \quad & r(\mathbf{y}, \mathbf{z}) && && (14a)\\
\text{s.t.} \quad & y_a \leq C_a && \forall a \in \mathcal{A} && (14b)\\
& y_a \geq 0 && \forall a \in \mathcal{A} && (14c)\\
& \mathbf{z} \in \mathcal{Z}(\mathbf{y}) && && (14d)
\end{aligned}$$

that is the penalty second-level Model (3) where we replace the objective function with $r(\mathbf{y}, \mathbf{z})$. Note that the feasibility region of this auxiliary model is $\mathcal{P}$. Let ξ be the value of the optimal solution of this auxiliary problem. Then ξ is a lower bound on the variable σ in Model (6). Therefore, it holds that

$$\xi \leq \overline{s} - \theta_{p^*}(x)\overline{g}_{p^*} - \sum_{a \in \mathcal{A}} \overline{k}_a x_a \leq \sigma$$

After resolving the definition of $\theta_{p^*}(x) = c_{p^*} + \sum_{a \in \mathcal{A}} M_a(1 - x_a) y_a^{p^*}$, this implies that $(\overline{s} - c_{p^*}\overline{g}_{p^*} - \xi)$ is a valid upper bound on all coefficients in the optimality cuts, as $\sigma \geq 0$. Hence, if a coefficient exceeds this bound, we can trim these coefficients to $(\overline{s} - c_{p^*}\overline{g}_{p^*} - \xi)$ without losing optimality. Note that when solving the Benders reduced subproblem Model (8), we do not require the M_a-coefficients, as it holds that $\overline{x}_a = 1$ for all $a \in p^*$.

5 Multi-Cut Benders Approach for Bilevel Network Design with Routing

If the second level decomposes into multiple independent subproblems, our Hierarchical Decomposition can be easily implemented within a multi-cut Benders framework. This structure commonly appears, e.g., in many network design problems, where the first level designs a transport network and the second level involves multiple users selecting paths within it. Here, we present a highly efficient implementation for the Hazmat Network Design Problem with Capacity Constraints (HNDPwCC), which generalizes numerous problems where paths are subject to capacity

constraints, while also demonstrating the necessary adjustments to extend our approach to a multi-cut Benders setting and integrate various speed-up techniques.

The classical Hazmat Network Design Problem (HNDP) is a well-studied, NP-hard bilevel problem (Amaldi et al., 2011; Fontaine & Minner, 2018; Kara & Verter, 2004) defined on a network graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ where the first level makes binary decisions if an arc $a \in \mathcal{A}$ is contained in the network ($x_a = 1$) or not ($x_a = 0$). The second level then consists of multiple truck drivers $u \in \mathcal{U}$, each selecting the shortest path in the resulting network from a given origin to their destination. The truck drivers select their path based on arc-cost $c_a^u \geq 0$, while the objective of the first level is to minimize the risk $r_a^u \geq 0$ of the selected paths.

In the HNDPwCC, we extend this problem setting to truck drivers that solve the Constrained Shortest Path Problem (CSP). For each arc a, we define additional resources l_a^u for each truck driver $u \in \mathcal{U}$ and restrict their consumption on a path to Q^u. The HNDPwCC is then formally formulated as a bilevel problem where the first level is defined as

$$\min \quad \sum_{u \in \mathcal{U}} \sum_{a \in \mathcal{A}} r_a^u y_a^u \tag{15a}$$

$$\text{s.t.} \quad x_a \in \{0,1\} \qquad \forall a \in \mathcal{A} \tag{15b}$$

$$\mathbf{y} \in \mathcal{S}(\mathbf{x}) \tag{15c}$$

while the truck routes given as arc-flows $\mathbf{y}$ are determined by an (optimistic) optimal solution of the second level

$$\min \quad \sum_{u \in \mathcal{U}} \sum_{a \in \mathcal{A}} c_a^u y_a^u \tag{16a}$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(i)} y_a^u - \sum_{a \in \delta^-(i)} y_a^u = d_i^u \qquad \forall u \in \mathcal{U}, i \in \mathcal{N} \tag{16b}$$

$$y_a^u \leq x_a \qquad \forall u \in \mathcal{U}, a \in \mathcal{A} \tag{16c}$$

$$\sum_{a \in \mathcal{A}} l_a^u y_a^u \leq Q^u \qquad \forall u \in \mathcal{U} \tag{16d}$$

$$y_a^u \in \{0,1\} \qquad \forall u \in \mathcal{U}, a \in \mathcal{A} \tag{16e}$$

Constraints (16b) are the flow conservation constraints, where d_i^u is an auxiliary parameter that is 1 for the origin and -1 for the destination of truck driver $u \in \mathcal{U}$. Constraints (16c) link the first and second level decisions, and capacity constraints (16d) ensure that each path satisfies the resource limit Q^u.

The extension to the CSP allows us to model more realistic routing decisions of the truck drivers. For example, l_a^u can represent travel time or energy consumption of electric trucks, and

Q^u then restricts the routes to a fixed time window or battery capacity, respectively. However, as the CSP is NP-hard, the existence of a compact LP is highly unlikely as it would lead to the polynomial hierarchy collapsing, making the classical approaches for HNDP based on strong duality not applicable to this setting.

Hence, we apply the Dantzig-Wolfe-like decomposition on the HNDPwCC. The master problem corresponds to the first-level Model (15). However, in our initial experiments, we observed that this formulation results in a high number of feasibility cuts being generated, as the master problem lacks any information that the x_a variables represent a network that should connect origin-destination pairs for truck drivers. Motivated by the partial Benders decomposition approach in Fontaine and Minner (2018), we include shortest path constraints into the master problem but relax integrality constraints on the y_a^u, resulting in

$$\min \quad \sum_{u \in \mathcal{U}} \sigma^u \tag{17a}$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(i)} y_a^u - \sum_{a \in \delta^-(i)} y_i^u = d_i^u \qquad \forall u \in \mathcal{U}, \forall i \in \mathcal{N} \tag{17b}$$

$$y_a^u \leq x_a \qquad \forall u \in \mathcal{U}, a \in \mathcal{A} \tag{17c}$$

$$\sum_{a \in \mathcal{A}} r_a^u y_a^u = \sigma^u \qquad \forall u \in \mathcal{U} \tag{17d}$$

$$x_a \in \{0,1\} \qquad \forall a \in \mathcal{A} \tag{17e}$$

$$\mathbf{y} \geq 0 \tag{17f}$$

$$\sigma^u \geq 0 \tag{17g}$$

The advantage of this new master formulation is that we provide the solver with additional information on the structure of the solution space while omitting the challenging resource constraints. Therefore, feasibility cuts are no longer needed to ensure connectivity in the network but only to ensure that each truck driver can select a resource-feasible path.

For the separation of the Dantzig-Wolfe-like cuts, observe that the second level Model (16) decomposes into $|\mathcal{U}|$ independent CSPs. Therefore, we generate Dantzig-Wolfe-like optimality and feasibility cuts for each of the subproblems independently, resulting in a higher number, but also stronger cuts. For the optimality cuts, we use variables σ^u to represent the optimal solution of truck driver u, and constraints (17d) link these with the included y_a^u-variables to strengthen the master problem relaxation.

For each truck driver u, we define $\overline{\mathcal{P}}^u$ as the set of all resource-feasible paths and solve the following LP to obtain the Dantzig-Wolfe cuts for the current master solution $\overline{x}_a$:

$$\max \quad s^u - \sum_{a \in \mathcal{A}} \overline{x}_a k^u_a - \theta_{p^*}(\overline{x}) g^u_{p^*} \tag{18a}$$

$$\text{s.t.} \quad s^u - \sum_{a \in \mathcal{A}_p} k^u_a - c^u_p g^u_{p^*} \leq r^u_p \qquad \forall p \in \overline{\mathcal{P}}^u \tag{18b}$$

$$s^u \text{ free} \tag{18c}$$

$$k^u_a \geq 0 \qquad \forall a \in \mathcal{A} \tag{18d}$$

$$g^u_{p^*} \geq 0 \tag{18e}$$

From Theorem 1, we know that the separation problem for constraints (18b) corresponds to solving the CSP with the new objective function $r^u_a + c^u_a \overline{g}_{p^*} + \overline{k}_a$. As both c^u_p and r^u_a are non-negative, we can solve it efficiently with an exact labeling algorithm/A^*-search for CSP (cf. Boland et al., 2006).

Model (18) contains multiple equivalent optimal solutions that result in cuts of highly varying quality. As we are highly interested in optimal solutions that result in cuts with low coefficient values, we apply a heuristic approach to generate Pareto-Optimal Benders cuts (cf. Rahmaniani et al., 2017). After solving Model (18) and obtaining an optimal solution value δ^u, we solve the following auxiliary LP:

$$\min \quad \sum_{a \in \mathcal{A}} k^u_a + \max_{a \in \mathcal{A}} \{M^u_a\} g^u_{p^*} \tag{19a}$$

$$\text{s.t.} \quad s^u - \sum_{a \in \mathcal{A}_p} k^u_a - c^u_p g^u_{p^*} \leq r^u_p \qquad \forall p \in \overline{\mathcal{P}}^u \tag{19b}$$

$$s^u - \theta_{p^*}(\overline{x}) g^u_{p^*} - \sum_{a \in \mathcal{A}} \overline{x}_a k^u_a = \delta^u \tag{19c}$$

$$s^u \text{ free} \tag{19d}$$

$$k^u_a \geq 0 \qquad \forall a \in \mathcal{A} \tag{19e}$$

$$g^u_{p^*} \geq 0 \tag{19f}$$

Constraint (19c) ensures that the found new solution remains optimal, while the objective function (19a) aims to minimize the coefficients in the resulting cut.

Because we generate individual cuts for each truck driver u, we cannot directly apply the approach from Section 4.4 to improve the big M induced by the reformulation. Instead, we solve the CSP for each u on the original network $\mathcal{G}$ with the objective to minimize risk. This risk-minimal solution is a natural lower bound on σ^u that we use to prune the cut coefficients.

6 Computational Results

To demonstrate the computational efficacy of our Hierarchical Decomposition approach, we implemented the presented framework for the HNDPwCC and compared it with the Benders-like approach (Israeli, 1999; Wood, 2011). Our Hierarchical Decomposition implementation includes the presented multi-cut Benders framework, including the warm start procedure from Remark 4, partial Benders decomposition, Pareto-Optimal Benders cuts, and the improved cut coefficients. The used Benders-like cuts model and details on the implementation are provided in Appendix C.

6.1 Test Instances

For our experiments, we extended the Sioux-Falls test instances with 24 nodes and 76 arcs from Fontaine and Minner (2018) to our setting by generating all necessary arc parameters randomly between 1 and 100 (same for all truck drivers) and generated 350 random origin-destination pairs, each representing an individual truck driver.

To test different parameters Q^u, we compute a lower bound Q^u_{min} corresponding to the minimal amount of resources any feasible path for u requires. Through preliminary experiments, we also find that $Q^u_{max} = 240$ is a good estimate where most (realistic) paths in the network are available. Based on this, we define $Q^u = Q^u_{min} + \alpha(Q^u_{max} - Q^u_{min})$, and report results for varying values of $\alpha \in [0, 1]$. Additionally, we consider the option *no restriction*, where we set $Q^u = 2400$, which is a sufficiently large value to ensure that no resource restrictions are enforced. The *no restriction* setting therefore corresponds to the classical HNDP.

6.2 Computational Environment

All tests were implemented in Java and executed on a virtual machine with an AMD EPYC 9334 32-Core 2.70 GHz processor and 62.5 GB RAM. A time limit of 3600 seconds was used. Gurobi was used to solve any incurring MIPs and LPs, while the CSP was solved by a basic implementation (without parallelization) of an A^*-search algorithm. No parallelization was used in the separation; i.e., in the separation step, the subproblems for each truck driver were solved sequentially. For both Hierarchical Decomposition and Benders-like implementations, an initial solution where each truck driver solves the CSP on the Sioux-Falls network was provided to Gurobi. The used Java code is publicly available at Zenodo (https://doi.org/10.5281/zenodo.14253685).

6.3 Benders-like vs. Hierarchical Decomposition

Table 2 reports our computational results for increasing α in 0.1 steps. For each scenario, 10 instances were generated, and we report the average results over these 10 runs, while the detailed results for each individual run can be found on Zenodo. We report the average runtime, size of the Branch&Bound tree, time spent in the separation loop, and the overall number of times we generated a new cut, for both Hierarchical Decomposition and Benders-like decomposition. Additionally, Figure 3 shows the relative improvement of our Hierarchical Decomposi-

tion compared to the Benders-like approach, i.e., $\frac{\text{Runtime Benders-like}}{\text{Runtime Hierarchical Decomposition}}$ for the runtime and $\frac{\text{\#Nodes Benders-like}}{\text{\#Nodes Hierarchical Decomposition}}$ for the number of generated Branch&Bound nodes.

	Hierarchical Decomposition				Benders-like Decomposition			
α values	Runtime	B&B Nodes	Separation Time	#Cuts	Runtime	B&B Nodes	Separation Time	#Cuts
0	11,0	239	4,00	6230	**2,8**	165	0,37	3395
0,1	15,0	576	5,20	7805	**7,3**	1009	0,52	4690
0,2	19,3	871	6,70	10010	**13,3**	2540	0,72	6405
0,3	**22,6**	1211	6,96	10255	57,4	5728	0,90	8050
0,4	**33,4**	2037	9,58	13895	147,5	15981	1,19	10640
0,5	**39,0**	1921	9,77	13860	378,9	28663	1,37	12075
0,6	**48,9**	3108	11,04	15750	781, 9*	53119	1,46	12985
0,7	**48,3**	2830	10,88	15050	636,7	39946	1,55	13790
0,8	**59,3**	3358	9,92	13755	774, 4*	35092	1,86	16765
0,9	**60,9**	3579	11,14	15190	706, 6*	32088	1,76	15995
1	**50,1**	2583	9,41	13055	761, 6*	32499	1,66	14840
n.r.	**38,9**	1859	9,18	12390	488,4	17519	1,58	13895

Figure 2: Computational performance of Benders-like and Hierarchical Decomposition. The *n.r.* stands for *no restriction*. All runtimes are given in seconds. *: Only 9 of the 10 instances were solved to optimality by the Benders-like decomposition within the time limit.

The results clearly show that our new approach is superior to the Benders-like cuts for challenging instances with $\alpha \geq 0.3$. The main advantages come from the ability to project out the integrality restrictions on the second-level variables from the master problem, resulting in significantly smaller Branch&Bound trees. For Benders-like cuts, Gurobi is forced to extensively branch on the second-level variables, which is highly ineffective. For the most challenging instances, we achieve an improvement in runtime by a factor of 14 and generate Branch&Bound trees that are over ten times smaller on average. However, it is also important to highlight that the Hierarchical Decomposition significantly profits from being directly linked to the classical Benders framework, allowing us to fall back on a rich literature base of speed-up techniques like Pareto-Optimal Benders cuts that significantly improve performance.

For $\alpha \leq 0.2$, the Benders-like cuts outperform our Hierarchical Decomposition. The main reason is the time spent in the separation loop is much higher for the Hierarchical Decomposition, which is expected as we need to solve an additional LP Model (18) for each truck driver, involving solving the CSP multiple times. For Hierarchical Decomposition, separation takes up around 10% of the overall runtime, while it is negligible for Benders-like cuts. Nonetheless, the computational benefits obtained from the Hierarchical Decomposition clearly outweigh the drawback of longer separation times for challenging instances. Also, implementing parallelization in the separation loop should strongly improve the separation of Hierarchical Decomposition if even better computational performance is required.

The somewhat surprising result to us was the *no restriction* case. Despite the capacity parameter Q^u being sufficiently large to invalidate the necessity to force the second-level variables to be integer, Gurobi consequently fails to detect this and extensively branches on the second-level variables. The only approach that successfully prevented Gurobi from branching was to remove the capacity constraints from the Benders-like formulation, which we do not consider a fair com-

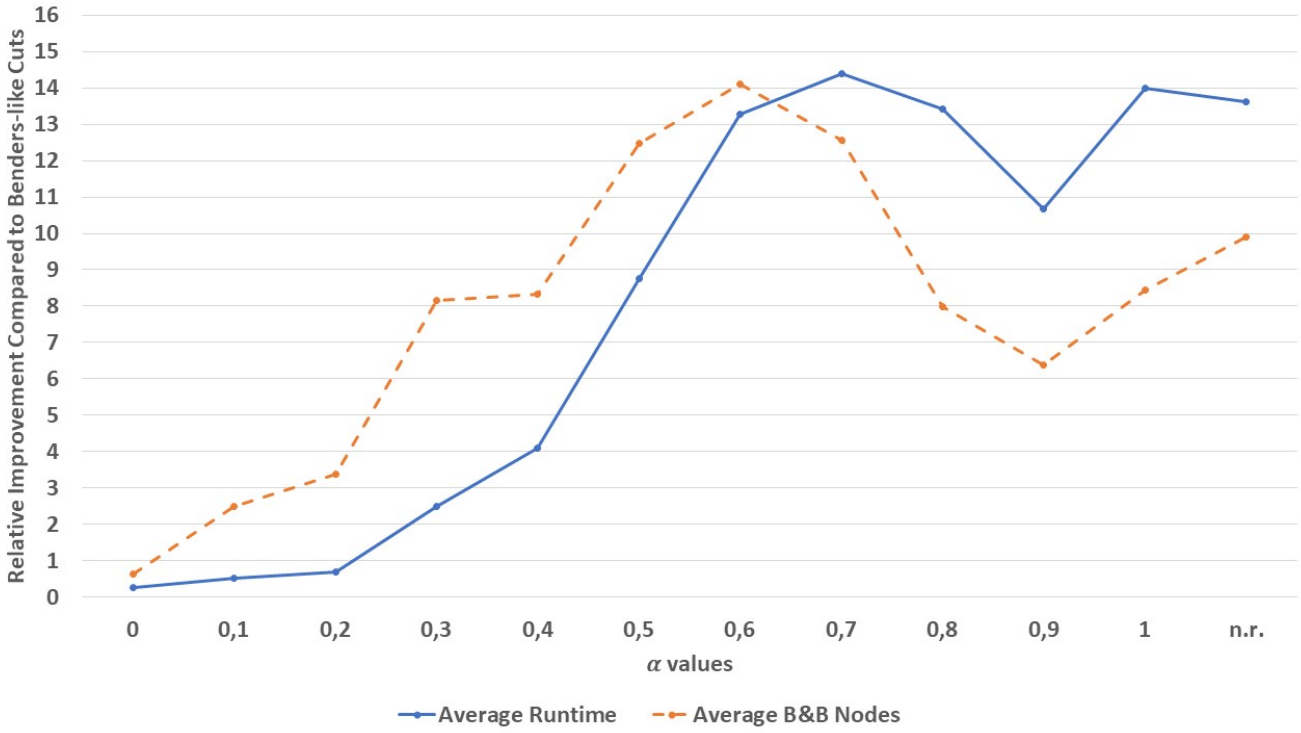

Figure 3: The relative improvements for runtime and the reduction in the number of Branch&Bound nodes of our Hierarchical Decomposition compared to Benders-like decomposition. The *n.r.* stands for *no restriction*.

parison. If it is already known that the problem reduces to the HNDP, we would employ the highly specialized algorithm from Fontaine and Minner (2018), and not Benders-like decomposition. However, these results indicate that our Hierarchical Decomposition can more easily deal with redundant information in the second-level formulation.

As a final remark, we would like to mention that we also conducted some experiments where we added Hierarchical Decomposition optimality cuts to the Benders-like cuts model. While this did not lead to better computational performance in most cases—in fact, the additional separation time sometimes worsened runtimes—we clearly observed that adding Hierarchical Decomposition optimality cuts improved the LP bound. This provides experimental proof that applying the Dantzig-Wolfe aggregation step strengthens the resulting LP relaxation and that Hierarchical Decomposition are based on a stronger LP formulation than the initial Benders-like cut model.

7 Conclusion

In this paper, we present the Hierarchical Decomposition for bilevel problems with binary linking variables. This novel decomposition approach partitions these bilevel problems into a sequence of MIP models where the subsequent model is used as a separation oracle for the previous model. These MIPs strongly preserve the original structure of the first- and second-level problems, enabling the application of specialized algorithms.

We present an efficient implementation of our framework for the HNDPwCC, demonstrating how it can be adjusted to the case with multiple independent second-level problems, and how speed-up techniques from classical Benders decomposition can be transferred to our framework. Our computational results clearly show that the new framework provides significant runtime benefits compared to Benders-like cuts.

While our computational results look promising, it is also important to highlight that the structure of HNDPwCC is highly beneficial to decomposition approaches like Benders-like cuts or our Hierarchical Decomposition. As a next step, we would like to extend our computational experiments to a wider range of bilevel problems with varying properties From a theoretical point of view, the question arises as to what extent the assumption of binary linking variables constraints is truly binding, or if it could be extended to integer or even linear first-level variables. Also, we address the optimistic bilevel case in this work. Future work should analyze whether it is possible to extend our approach to the pessimistic case.

References

Amaldi, E., Bruglieri, M., & Fortz, B. (2011). On the Hazmat Transport Network Design Problem. *Lecture Notes in Computer Science*, 327–338. DOI: 10.1007/978-3-642-21527-8_38.

Arslan, O., Jabali, O., & Laporte, G. (2018). Exact Solution of the Evasive Flow Capturing Problem. *Operations Research*, *66*(6), 1625–1640. DOI: 10.1287/opre.2018.1756.

Avraamidou, S., & Pistikopoulos, E. N. (2019). B-pop: Bi-level parametric optimization toolbox. *Computers & Chemical Engineering*, *122*, 193–202. DOI: 10.1016/j.compchemeng.2018.07.007.

Bazgan, C., Toubaline, S., & Vanderpooten, D. (2012). Critical edges/nodes for the minimum spanning tree problem: Complexity and approximation. *Journal of Combinatorial Optimization*, *26*(1), 178–189. DOI: 10.1007/s10878-011-9449-4.

Boland, N., Dethridge, J., & Dumitrescu, I. (2006). Accelerated label setting algorithms for the elementary resource constrained shortest path problem. *Operations Research Letters*, *34*(1), 58–68. DOI: 10.1016/j.orl.2004.11.011.

Byeon, G., & Van Hentenryck, P. (2022). Benders subproblem decomposition for bilevel problems with convex follower. *INFORMS Journal on Computing*, *34*(3), 1749–1767. DOI: 10.1287/ijoc.2021.1128.

Caprara, A., Carvalho, M., Lodi, A., & Woeginger, G. J. (2016). Bilevel knapsack with interdiction constraints. *INFORMS Journal on Computing*, *28*(2), 319–333. DOI: 10.1287/ijoc.2015.0676.

Cerulli, M., Archetti, C., Fernández, E., & Ljubić, I. (2024). A bilevel approach for compensation and routing decisions in last-mile delivery. *Transportation Science*. DOI: 10.1287/trsc.2023.0129.

Chouman, M., Crainic, T. G., & Gendron, B. (2017). Commodity Representations and Cut-Set-Based Inequalities for Multicommodity Capacitated Fixed-Charge Network Design. *Transportation Science*, *51*(2), 650–667. DOI: 10.1287/trsc.2015.0665.

Clautiaux, F., & Ljubić, I. (2024). Last fifty years of integer linear programming: A focus on recent practical advances. *European Journal of Operational Research*. DOI: 10.1016/j.ejor.2024.11.018.

Codato, G., & Fischetti, M. (2006). Combinatorial benders' cuts for mixed-integer linear programming. *Operations Research*, *54*(4), 756–766. DOI: 10.1287/opre.1060.0286.

Della Croce, F., & Scatamacchia, R. (2020). An exact approach for the bilevel knapsack problem with interdiction constraints and extensions. *Mathematical Programming*, *183*(1–2), 249–281. DOI: 10.1007/s10107-020-01482-5.

Desrosiers, J., & Lübbecke, M. E. (2006). A primer in column generation. In *Column Generation* (pp. pp. 1–32) (pp. 1–32). Springer-Verlag. DOI: 10.1007/0-387-25486-2_1.

Fischetti, M., Ljubić, I., Monaci, M., & Sinnl, M. (2017). A new general-purpose algorithm for mixed-integer bilevel linear programs. *Operations Research*, *65*(6), 1615–1637. DOI: 10.1287/opre.2017.1650.

Fischetti, M., Ljubić, I., Monaci, M., & Sinnl, M. (2019). Interdiction games and monotonicity, with application to knapsack problems. *INFORMS Journal on Computing*, *31*(2), 390–410. DOI: 10.1287/ijoc.2018.0831.

Fischetti, M., Monaci, M., & Sinnl, M. (2018). A dynamic reformulation heuristic for generalized interdiction problems. *European Journal of Operational Research*, *267*(1), 40–51. DOI: 10.1016/j.ejor.2017.11.043.

Fontaine, P., & Minner, S. (2014). Benders Decomposition for Discrete–Continuous Linear Bilevel Problems with application to traffic network design. *Transportation Research Part B: Methodological*, *70*, 163–172. DOI: 10.1016/j.trb.2014.09.007.

Fontaine, P., & Minner, S. (2016). A dynamic discrete network design problem for maintenance planning in traffic networks. *Annals of Operations Research*, *253*(2), 757–772. DOI: 10.1007/s10479-016-2171-y.

Fontaine, P., & Minner, S. (2018). Benders decomposition for the Hazmat Transport Network Design Problem. *European Journal of Operational Research*, *267*(3), 996–1002. DOI: 10.1016/j.ejor.2017.12.042.

Furini, F., Ljubić, I., Martin, S., & San Segundo, P. (2019). The maximum clique interdiction problem. *European Journal of Operational Research*, *277*(1), 112–127. DOI: 10.1016/j.ejor.2019.02.028.

Gao, Z., Wu, J., & Sun, H. (2005). Solution algorithm for the bi-level discrete network design problem. *Transportation Research Part B: Methodological*, *39*(6), 479–495. DOI: 10.1016/j.trb.2004.06.004.

Israeli, E. (1999). *System interdiction and defense* [Doctoral dissertation, Naval Postgraduate School].

Israeli, E., & Wood, R. K. (2002). Shortest-path network interdiction. *Networks*, *40*(2), 97–111. DOI: 10.1002/net.10039.

Jeroslow, R. G. (1985). The polynomial hierarchy and a simple model for competitive analysis. *Mathematical Programming*, *32*(2), 146–164. DOI: 10.1007/bf01586088.

Kara, B. Y., & Verter, V. (2004). Designing a road network for hazardous materials transportation. *Transportation Science*, *38*(2), 188–196. DOI: 10.1287/trsc.1030.0065.

Kleinert, T., Labbé, M., Ljubić, I., & Schmidt, M. (2021). A survey on mixed-integer programming techniques in bilevel optimization. *EURO Journal on Computational Optimization*, *9*, 100007. DOI: 10.1016/j.ejco.2021.100007.

Kleinert, T., Labbé, M., Plein, F., & Schmidt, M. (2020). Technical note—there's no free lunch: On the hardness of choosing a correct big-m in bilevel optimization. *Operations Research*, *68*(6), 1716–1721. DOI: 10.1287/opre.2019.1944.

Kleinert, T., & Schmidt, M. (2023). Why there is no need to use a big-m in linear bilevel optimization: A computational study of two ready-to-use approaches. *Computational Management Science*, *20*(1). DOI: 10.1007/s10287-023-00435-5.

Kleniati, P.-M., & Adjiman, C. S. (2015). A generalization of the branch-and-sandwich algorithm: From continuous to mixed-integer nonlinear bilevel problems. *Computers and Chemical Engineering*, *72*, 373–386. DOI: 10.1016/j.compchemeng.2014.06.004.

Ley, E., & Merkert, M. (2024). Solution methods for partial inverse combinatorial optimization problems in which weights can only be increased. https://optimization-online.org/?p=25609 Accessed.

Lim, C., & Smith, J. C. (2007). Algorithms for discrete and continuous multicommodity flow network interdiction problems. *IIE Transactions*, *39*(1), 15–26. DOI: 10.1080/07408170600729192.

Mahdavi Pajouh, F. (2019). Minimum cost edge blocker clique problem. *Annals of Operations Research*, *294*(1–2), 345–376. DOI: 10.1007/s10479-019-03315-x.

Marković, N., Ryzhov, I. O., & Schonfeld, P. (2014). Evasive flow capture: Optimal location of weigh-in-motion systems, tollbooths, and security checkpoints. *Networks*, *65*(1), 22–42. DOI: 10.1002/net.21581.

Marković, N., Ryzhov, I. O., & Schonfeld, P. (2017). Evasive flow capture: A multi-period stochastic facility location problem with independent demand. *European Journal of Operational Research*, *257*(2), 687–703. DOI: 10.1016/j.ejor.2016.08.020.

Mattia, S. (2024). Reformulations and complexity of the clique interdiction problem by graph mapping. *Discrete Applied Mathematics*, *354*, 48–57. DOI: 10.1016/j.dam.2021.06.008.

Moore, J. T., & Bard, J. F. (1990). The mixed integer linear bilevel programming problem. *Operations Research*, *38*(5), 911–921. DOI: 10.1287/opre.38.5.911.

Pemmaraju, S., & Skiena, S. (2003). *Computational discrete mathematics: Combinatorics and graph theory with mathematica ®*. Cambridge University Press. DOI: https://doi.org/10.1017/CBO9781139164849.

Pineda, S., & Morales, J. M. (2019). Solving linear bilevel problems using big-ms: Not all that glitters is gold. *IEEE Transactions on Power Systems*, *34*(3), 2469–2471. DOI: 10.1109/tpwrs.2019.2892607.

Poirion, P.-L., Toubaline, S., D'Ambrosio, C., & Liberti, L. (2020). Algorithms and applications for a class of bilevel milps. *Discrete Applied Mathematics*, *272*, 75–89. DOI: 10.1016/j.dam.2018.02.015.

Rahmaniani, R., Crainic, T. G., Gendreau, M., & Rei, W. (2017). The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, *259*(3), 801–817. DOI: 10.1016/j.ejor.2016.12.005.

Tahernejad, S., Ralphs, T. K., & DeNegre, S. T. (2020). A branch-and-cut algorithm for mixed integer bilevel linear optimization problems and its implementation. *Mathematical Programming Computation*, *12*(4), 529–568. DOI: 10.1007/s12532-020-00183-6.

Taninmis, K., & Sinnl, M. (2022). A branch-and-cut algorithm for submodular interdiction games. *INFORMS Journal on Computing*, *34*(5), 2634–2657. DOI: 10.1287/ijoc.2022.1196.

Tavaslıoğlu, O., Prokopyev, O. A., & Schaefer, A. J. (2019). Solving stochastic and bilevel mixed-integer programs via a generalized value function. *Operations Research*, *67*(6), 1659–1677. DOI: 10.1287/opre.2019.1842.

Wang, L., & Xu, P. (2017). The watermelon algorithm for the bilevel integer linear programming problem. *SIAM Journal on Optimization*, *27*(3), 1403–1430. DOI: 10.1137/15m1051592.

Wood, R. K. (2011, January). Bilevel network interdiction models: Formulations and solutions. DOI: 10.1002/9780470400531.eorms0932.

Xu, P., & Wang, L. (2014). An exact algorithm for the bilevel mixed integer linear programming problem under three simplifying assumptions. *Computers & Operations Research*, *41*, 309–318. DOI: 10.1016/j.cor.2013.07.016.

A General Structure of Second Level

Here, we shortly want to address Remark 1 and demonstrate how an arbitrary second level problem given as $\min_{z \in \mathcal{Z}} \{\mathbf{c_z}^T \mathbf{z}\}$ with $\mathcal{Z} = \{\mathbf{V_x x} + \mathbf{V_z z} \geq \mathbf{v}; \mathbf{z} \in \mathbb{R}^{n_z} \times \mathbb{N}^{m_z}\}$ can be transformed into the interdiction structure used in Model (2). To this end, we define an auxiliary variable $y_a \geq 0$, $a \in \mathcal{A}$, and substitute the constraints in $\mathcal{Z}$ with

$$\begin{aligned} \mathbf{V_x y} + \mathbf{V_z z} &\geq \mathbf{v} \\ y_a &\leq x_a && \forall a \in \mathcal{A} \\ 1 - y_a &\leq 1 - x_a && \forall a \in \mathcal{A} \end{aligned}$$

where the last two constraints enforce $y_a = x_a$ for all $a \in \mathcal{A}$. By substituting ${y_a}' = 1 - y_a$ and $x' = 1 - x_a$ with additional auxiliary variables ${y_a}'$ and ${x_a}'$, we obtain the structure from Model (2).

B Dealing with Coupling Constraints

In this section, we shortly discuss to which extend coupling constraints can be integrated into our framework. Hence, assume that the first level is extended by additional coupling constraints

$$\mathbf{Q_h h} + \mathbf{Q_x x} + \mathbf{Q_y y} \geq \mathbf{q}.$$

Note that we can disregard the second level variables $\mathbf{z}$ following Remark 1.

If we employ our Dantzig-Wolfe decomposition approach from Section 4.1, we obtain the following reformulation including coupling constraints

$$\begin{aligned}
\min \quad & r(\mathbf{h}, \mathbf{x}) + \sum_{p \in \overline{\mathcal{P}}} r_p f_p && && \text{(20a)}\\
\text{s.t.} \quad & \mathbf{Q_h h} + \mathbf{Q_x x} + \mathbf{Q_y y} \geq \mathbf{q} && && \text{(20b)}\\
& y_a = \sum_{p \in \overline{\mathcal{P}}} y_a^p f_p && \forall a \in \mathcal{A} && \text{(20c)}\\
& \sum_{p \in \overline{\mathcal{P}}} f_p = 1 && && \text{(20d)}\\
& \sum_{p \in \overline{\mathcal{P}}_a} f_p \leq C_a x_a && \forall a \in \mathcal{A} && \text{(20e)}\\
& \sum_{p' \in \overline{\mathcal{P}}} c_{p'} f_{p'} \leq c_p + \sum_{a \in \mathcal{A}} M_a (1 - x_a) y_a^p && \forall p \in \overline{\mathcal{P}} && \text{(20f)}\\
& f_p \in \{0; 1\} && \forall p \in \overline{\mathcal{P}} && \text{(20g)}\\
& \mathbf{h} \in \mathcal{H}(\mathbf{x}) && && \text{(20h)}\\
& x_a \in \{0, 1\} && \forall a \in \mathcal{A} && \text{(20i)}
\end{aligned}$$

The main addition are the coupling constraints (20b) and constraints (20c) that link the y_a-variables with their aggregated f_p-variables counterpart. Merging the constraints (20b) and (20c) results in

$$\mathbf{Q_h h} + \mathbf{Q_x x} + \mathbf{Q_f f} \geq \mathbf{q} \tag{21}$$

for a fitting coefficient matrix $\mathbf{Q_f}$. Constraints (21) introduce a new complexity into the Dantzig-Wolfe reformulated model (20), as the inner optimization problem $\min_{\mathbf{f}} \left\{ \sum_{p \in \overline{\mathcal{P}}} r_p f_p \mid (20c) - (20f); (21); f_p \in \{0, 1\} \, \forall p \in \overline{\mathcal{P}} \right\}$ is no longer necessarily an LP. Before applying the Benders decomposition approach presented in Section 4.2, we therefore have to first check if the integrality conditions on the f_p-variables can be relaxed. If this is not the case, we could still solve Model (20) with a Branch-Price-and-Cut approach, but the computational challenges involved would most likely make it inferior to other solution methods.

Nonetheless, there are many types of coupling constraints that do preserve the LP structure of the inner optimization problem. For example, if the second-level variables $\mathbf{y}$ are binary, resource requirement constraints $y_a = 1$ for all $a \in \mathcal{A}' \subseteq \mathcal{A}$, e.g., often found in partial inverse optimization problems (Ley & Merkert, 2024), result in aggregated constraints $\sum_{p \in \overline{\mathcal{P}}_a} f_a = 1$ for all $a \in \mathcal{A}'$ that preserve this property. In summary, integrating coupling constraints into our approach requires us to study the structure of the resulting inner optimization problem, making them hard to integrate into an automated framework.

C Bender-like Cuts Model for the HNDPwCC

For the Benders-like cuts approach to HNDPwCC presented in Section 5, we implemented the following master MIP model

$$\min \quad \sum_{u\in\mathcal{U}}\sum_{a\in\mathcal{A}} r_a^u y_a^u \tag{22a}$$

$$\text{s.t.} \quad \sum_{a\in\mathcal{A}}(1-x_a) \le B \tag{22b}$$

$$\sum_{a\in\delta^+(i)} y_a^u - \sum_{a\in\delta^-(i)} y_a^u = d_i^u \qquad \forall u\in\mathcal{U}, \forall i\in\mathcal{N} \tag{22c}$$

$$y_a^u \le x_a \qquad \forall u\in\mathcal{U}, a\in\mathcal{A} \tag{22d}$$

$$\sum_{a\in\mathcal{A}} l_a^u y_a^u \le Q^u \qquad \forall u\in\mathcal{U} \tag{22e}$$

$$y_a^u \in \{0,1\} \qquad \forall u\in\mathcal{U}, a\in\mathcal{A} \tag{22f}$$

$$x_a \in \{0,1\} \qquad \forall a\in\mathcal{A} \tag{22g}$$

We solve this model until all variables are integer, i.e, until a new MIP incumbent node is found. Then, we solve for each truck driver $u\in\mathcal{U}$ the CSP in the network induced by the found x-solution to find a path $p\in\overline{\mathcal{P}}^u$ that induces a violated Benders-like cut

$$\sum_{a\in\mathcal{A}} c_a^u y_a^u \le \sum_{a\in p} c_a^u + \sum_{a\in p} M_a^u (1-x_a).$$

If no violated cut is found, we terminate with the optimal solution.

The big M values M_a^u are naturally bounded by the length of the longest (according to cost c_a^u) path within the network (cf. Lim & Smith, 2007). As computing the simple longest path is strongly NP-hard, we instead use a heuristic to estimate its length. The heuristic computes the maximum spanning tree by negating the cost for each edge and applying Kruskal's algorithm (Pemmaraju & Skiena, 2003). The overall cost of the maximum spanning tree is then an upper bound on the length of the longest path, as otherwise extending the longest path to a spanning tree would result in a new spanning tree with higher cost, a contradiction.

In some initial parameter tuning, we observed that branching on the x_a variables results in better computational performance, as first fixing the network layout and then searching for the CSP seems to be computationally beneficial. Therefore, we set the branching priorities within Gurobi so that it first branches on the x_a variables, and only if all are integer on the y_a^u variables.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT to rewrite the original draft with the goal of improving readability and correct spelling/grammatical mistakes. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.