Binary Integer Program Reformulation: A Set System Approximation Approach

Ningji Wei

Department of Industrial, Manufacturing & Systems Engineering, Texas Tech University

Abstract

This paper presents a generic reformulation framework for binary integer programs (BIPs) that does not impose additional specifications on the objective function or constraints. To enable this generality, we introduce a set system approximation theory designed to identify the tightest inner and outer approximations for any binary solution space using special types of set systems. This approach leads to flexible reformulation strategies for BIPs, particularly focusing on the combination of set covering and subtour elimination inequalities. We provide an efficient separation subroutine based on a given membership oracle, and derive facet-defining conditions of these inequalities for general BIPs. We also explore the implications of this methodology on various problem instances, uncovering new solution strategies and structural insights for classic problems such as the longest path and traveling salesman. Additionally, we extend the structural aspect of max-flow min-cut theorem to a broader context of set system duality. Finally, we demonstrate the flexibility and efficiency of the proposed framework through a case study on a network site selection problem with distributionally robust chance constraints.

Keywords: Benders decomposition, set covering, subtour elimination

1 Introduction

Given an arbitrary binary solution space

$$\mathcal{X} := \{x \in \{0,1\}^n \mid g(x) \le 0\}$$

described by some general function $g : \mathbb{R}^n \to \mathbb{R}^m$, this paper explores a generic binary space representation theory that can exactly rewrite $\mathcal{X}$ into the following form (see derivation in Section 4)

$$\mathcal{X} = \left\{ x \in \{0,1\}^n \;\middle|\; \begin{array}{ll} \sum_{i \in T} x_i \ge z_k, & \forall k \in K, T \in \Omega_k^u \\ \sum_{i \in S} x_i \le |S| - z_k, & \forall k \in K, S \in \Omega_k^l \\ \sum_{k \in K} z_k = 1. & \end{array} \right\}, \tag{1}$$

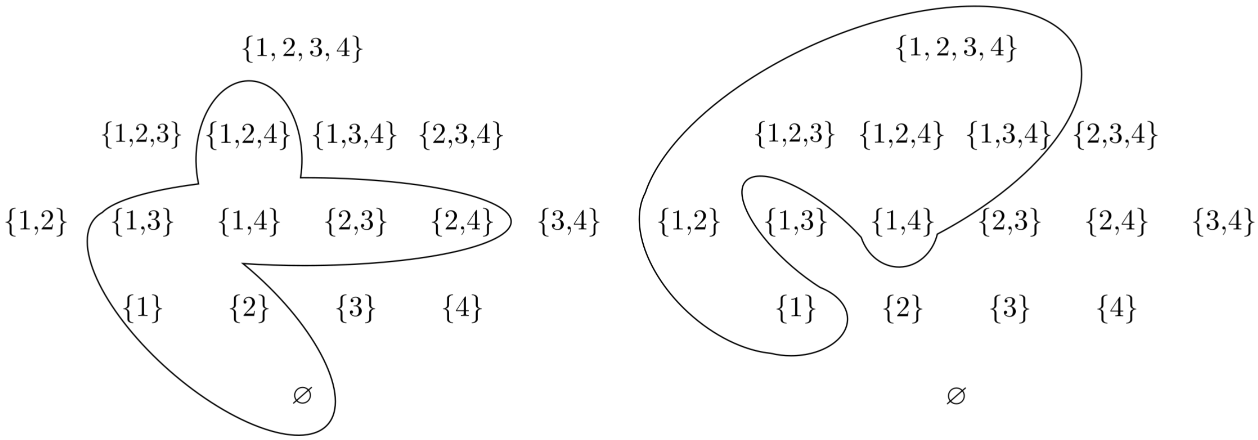

Figure 1: Two set systems, Ω_1 (left) and Ω_2 (right), each defined over the ground set $\Delta = \{1, 2, 3, 4\}$. Both set systems encompass all elements enclosed within their respective curves.

where $\{\Omega_k^u, \Omega_k^l\}_{k\in K}$ are special types of index sets for characterizing $\mathcal{X}$. This development offers flexible reformulation strategies for general *binary integer programs* (BIPs) of the form

$$\Pi : \min_{x\in\mathcal{X}:=\bigcap_{j\in J}\mathcal{X}_j} f(x), \tag{2}$$

as it allows for the reformulation of either the entire $\mathcal{X}$ or individual components $\mathcal{X}_j$ using the proposed representation, without imposing extra restrictions on the objective function f or the solution space $\mathcal{X}$. Due to such relaxed conditions, Formulation (2) can be used to describe a variety of interesting optimization problems, including but not limited to the following.

- Binary mixed integer programs with $\mathcal{Y}_x$ as the decision space of continuous variables,

$$\min_{x\in\mathcal{X}, y\in\mathcal{Y}_x} f(x, y).$$

- Multistage stochastic BIPs with ξ_i's as uncertain parameters

$$\min_{x_1\in\mathcal{X}_1} \mathbb{E}[f_1(x_1, \xi_1) + \min_{x_2\in\mathcal{X}_2} \mathbb{E}[f_2(x_2, \xi_2) + \cdots]].$$

- BIPs with distributionally robust chance constraints where $\mathfrak{P}$ is a set of distributions

$$\begin{aligned} &\min_{x\in\mathcal{X}} f(x) \\ &\text{s.t. } \min_{\mathbb{P}\in\mathfrak{P}} \mathbb{P}(g_i(x, \xi) \leq 0) \geq 1 - \epsilon, \ \forall i \in I. \end{aligned}$$

To achieve such generality, we explore a new analysis framework called the *set system approximation*. We identify $\{0, 1\}^n$ with the power set $\mathscr{P}(\Delta)$ for the ground set $\Delta = \{1, 2, \ldots, n\}$ and consider the solutions space $\mathcal{X}$ as a set system (a family of sets) embedded within $\mathscr{P}(\Delta)$ (see Figure 1). Then, we aim to develop a new theory that can approximate an arbitrary set system both internally and externally, using upper and lower set systems—the ones that contain either all the supersets or subsets of their members. This bears resemblance to convex analysis where "simple" functions such as convex and concave functions are exploited to approximate the general ones. This perspective also leads to the set system duality, generalizing the structural aspect of the classic max-flow min-cut theorem.

While our analytical approach differs from classic methodologies in the literature, the reformulation presented above reveals a strong connection with two well-established constraint types: the set covering inequalities [4] and the subtour elimination inequalities from the traveling salesman problem (TSP) [5]. Consequently, this reformulation method is also closely linked to the seminal works in combinatorial Bender's cuts [12] and logic-based Bender's decomposition [18].

1.1 Previous Works

Two special types of inequalities have been notably prevalent across various instance problems with distinct characteristics. Set covering inequalities are widely employed in different contexts, such as interdiction games [19, 26, 38], vehicle routing [30], network design [16], power grid optimization [39], and facilitate location problems [10], often enhancing the branch-and-cut implementation framework. Moreover, these covering inequalities have shown to possess strong facet properties in various problem settings under certain conditions [32, 33, 36]. Similarly, subtour elimination inequalities have become a critical reformulation component in transportation [1, 14, 21], production scheduling [11], location-routing problems [23], and have demonstrated significant impact in computational efficiency when properly strengthened and implemented [13, 34].

In many applications, these two types of inequalities are derived using Benders feasibility cuts [6], with strengthening steps to eliminate the big-M constants. This technique, known as *combinatorial Benders decomposition* (CBD), has subsequently been generalized to mixed integer linear programs with binary and continuous variables [12]. In a more general setting where the problem can be nonlinear, the logic-based Benders decomposition (LBBD) framework was introduced in the seminal work [18]. LBBD recognizes that, when solving a general optimization problem through a master-subproblem framework, new constraints for the master problem can be logically deduced from the value and feasibility of the subproblem, where such inference rules often need to be tailored to specific applications [17]. Due to its generality, this methodology has been widely applied in multiple domains, including transportation [7], production [25], supply chain management [29], and telecommunication [8], among others [9, 24, 40]. For a comprehensive coverage of this topic, we refer interested readers to the book [17] and the review paper [31] .

One universal type of inequality that has been identified in LBBD is termed the *nogood cuts*, defined as follows.

$$\sum_{i\in T} x_i + \sum_{i\in \Delta\setminus T} (1 - x_i) \geq 1.$$

That is, for a non-solution T, at least one value of its indicator vector must be changed to remove it from the solution space. A special type of nogood cut is the monotone cut, defined as

$$\sum_{i \in \Delta \setminus T} (1 - x_i) \geq 1 \Longleftrightarrow \sum_{i \in \Delta \setminus T} x_i \leq |\Delta \setminus T| - 1,$$

which is valid to use if the optimization problem has a monotone objective function. Since there is a similarity between the monotone cuts derived from the LBBD framework and the constraint sets in (1) derived using the proposed methodology, we provide a detailed comparison of these two methodologies in the next subsection.

1.2 Relation to LBBD

The major distinctions between LBBD and the proposed reformulation framework stem from the fundamental goals of these two methodologies.

In essence, LBBD was designed to generalize the classic Benders decomposition for linear programs to optimization problems in nonlinear settings, adopting the strategy of "learning from one's mistakes" [18]. Thus, the main focus of LBBD is to generate valid inequalities from the subproblem using logical inferences to refine the solution in the master problem. For instance, suppose the subproblem under x_T, the indicator vector of some structure $T \subseteq \Delta$ obtained from the master problem, is infeasible, a logical conclusion is to add the associated nogood cut into the master problem. Although the monotone cut is much stronger than the basic nogood cut, it can be introduced only if it is logically valid, such as when the objective function is monotone on x.

In comparison, the goal of the proposed reformulation framework is rooted in the binary space approximation and representation. Consequently, the main research question becomes how well an arbitrary binary space $\mathcal{X}$ can be approximated externally and internally using set covering and subtour elimination inequalities, without considering whether they are logically valid for the original problem. In particular, when serving as an inner approximation, the introduced constraints may not be valid.

Therefore, despite some similarities in certain facets of these two methodologies, the significant distinction in their design goals results in differences in multiple aspects, as outlined below.

- **Theoretical tools**: The fundamental theory in LBBD is the inference duality [17], which provides the tightest lower bound on the objective value using logical deduction on the constraint set. This extends the master-subproblem solution approach from the classic Benders decomposition to general settings without the aid of LP duality. In contrast, the proposed reformulation framework relies on set system approximation theory (Section 2), which derives the tightest inner and outer approximation of a binary space $\mathcal{X}$ using upper and lower set systems.

- **Reformulation focus**: Since the inference dual in LBBD is abstractly defined using the class of inequalities that can be deduced from the original problem's constraint set, its reformulation task mainly focuses on defining these logic-based constraints in specific problem settings. However, since the proposed methodology is rooted in set system approximation, reformulation (1) encourages decomposing any binary space $\mathcal{X}$ into the intersections and unions of upper and lower set systems (Section 4.2).

- **Separation focus**: Similar to Benders decomposition, constraint separation in LBBD has a dynamic here-and-now nature. That is, the separation subroutine aims to identify and strengthen a logic-based inequality from the incumbent solution obtained in the master problem iteratively. On the other hand, the proposed framework adopts a more static and structure-based perspective by providing a computation mechanism to globally identify the structures in Ω_k^u and Ω_k^l in (1) for a given binary space $\mathcal{X}$ (Examples 3, 4, and 5). Then, specific separation algorithms can be developed to generate these structures. Moreover, this structure-centric perspective leads to the theory of structural duality (Section 5), generalizing the structural aspect of the classic max-flow-min-cut theorem.

In conclusion, the proposed framework provides a new approximation and representation methodology for general binary spaces, encompassing theoretical tools, reformulation techniques, separation methods, and computational validations. Additionally, some results of this framework offer a new perspective for interpreting monotone cuts in the existing literature of CDB and LBBD.

1.3 Contributions

Our main contributions include the following.

- *Binary Space Representation Theory:* We develop a set system approximation theory that constructs the tightest inner and outer approximations for any given set system using upper and lower systems. To support this analysis, we establish a set of algebraic rules governing various set system operators. These approximation results are then translated into four inner and outer reformulations for general BIPs, with identified conditions for exactness. Additionally, we broaden the applicability of these reformulation techniques by analyzing the properties of the objective function and introducing a decomposition scheme for arbitrary set systems. Finally, these results lead to a dual correspondence for set systems, generalizing the structural aspect of classic flow-cut duality.

- *General Constraint Separation and Facet Analysis:* To support a cut generation framework for the proposed reformulation, we provide a general constraint separation subroutine based on any given solution membership oracle. We establish the corresponding complexity as $O(\log(n) \cdot \tau(\Omega))$, where n is the dimension of the solution space and $\tau(\Omega)$ represents the complexity of the oracle. Additionally, we extend facet analysis results from the set covering polytope literature to general BIPs, identifying if-and-only-if conditions for the proposed reformulation inequalities to be facet-defining.

- *Solution Strategies for Multiple BIPs:* We apply our framework to various BIP instances, including the shortest path, longest path, and traveling salesman problems, yielding new insights and solution strategies. For the longest path problem, we rediscover the strong claw and cycle-based inequalities from [22] using a different perspective, and introduce a new cut generation strategy based solely on elimination inequalities. Moreover, we conduct a case study on a network site selection problem with distributionally robust chance constraints, exploring four different solution strategies by applying the proposed reformulation method to different parts of the constraint set, highlighting the framework's flexibility. Our results show that, when properly implemented, the proposed reformulation method can significantly outperform the direct approach in terms of efficiency.

One benefit of the reformulation in (1) is that existing facet analysis and constraint strengthening methods for covering and elimination constraints can be directly applied to study various BIPs. This brings abundant valid and supervalid constraints that can potentially enhance existing solution methods. However, this paper does not focus on these facet analysis and strengthening techniques, as some general results have already been developed in the literature. For instance, facet-defining conditions for set covering inequalities are detailed in [3, 36], methods to generate valid inequalities from covering and packing constraints can be found in [2, 15, 20, 36, 41], and the technique called $\mathcal{P}$-structure separation introduced in [37] can be used to identify supervalid inequalities that could potentially eliminate feasible solutions without affecting optimality.

The rest of the paper is organized as follows. In Section 2, we establish the fundamental set system approximation theory. In Section 3, we delve into reformulation approximation techniques and assess their separation subroutines and runtime complexity. Section 4 introduces methods to extend the reformulation framework for arbitrary set systems. In Section 5, we generalize the classic flow-cut duality to arbitrary set systems and identify such structural dual pairs in various problem settings. Section 6 presents a case study on the network site selection problem with distributionally robust constraints, demonstrating the flexibility and efficiency offered by the proposed framework. Finally, Section 7 concludes with key remarks and summarize of our findings. To better streamline the paper, we defer all the proofs to Appendix A.

2 Set System Approximation

As previously mentioned, the central concept of this paper involves identifying a given binary solution space, denoted as $\mathcal{X}$, with its corresponding set system representation. Following this, we aim to develop a method for approximating set systems as a means to redefine the description of $\mathcal{X}$. This section is dedicated to establishing the theory for this set system approximation.

2.1 Preliminaries

In this subsection, we introduce several basic concepts, including set systems and their operators, for later development.

2.1.1 Set System

For a given binary decision space $\mathcal{X} \subseteq \{0,1\}^n$, we call $\Delta := \{1, 2, \ldots, n\}$ the ground set and the associated power set $\mathscr{P}(\Delta)$ the family of structures. Clearly, there is a bijective relation between elements in $\{0,1\}^n$ and structures in $\mathscr{P}(\Delta)$ described by $T_x := \{i \in \Delta \mid x_i = 1\}$. Then, every $\Omega \subseteq \mathscr{P}(\Delta)$ is termed a *set system*, i.e., a family of structures equipped with the natural ordering induced by the inclusion relation $\subseteq$. In particular, $\Omega_{\mathcal{X}} := \{T_x \mid x \in \mathcal{X}\}$ is the set system representation of $\mathcal{X}$. Conversely, $\mathcal{X}_\Omega$ denotes the vector representation of some given set system Ω. We use $\mathscr{P}^2(\Delta)$ to denote the family of all set systems defined on Δ.

2.1.2 Structure Operator & Set System Operator

A *structure operator* and a *set system operator* refer to any function f defined on $\mathscr{P}(\Delta)$ and $\mathscr{P}^2(\Delta)$, respectively. In particular, every structure operator f also induces a set system operator $\widehat{f}$ by

$$\widehat{f}(\Omega) = \{f(T) \mid T \in \Omega\}.$$

With a slight abuse of notation, we will use the same function notation for a structure operator and its induced set system operator (e.g., f is interpreted as $\widehat{f}$ when the input argument is a set system). Often, we will simply refer to both as operators, especially when their types are either apparent or inessential in the context. An operator is said to be *increasing* or *order-preserving* if $f(T) \subseteq f(T')$ whenever $T \subseteq T'$, and is called *decreasing* or *order-reversing* for the opposite case.

Two mathematical statements regarding set systems are called *dual statements* if one can be converted to the other by the following substitution rules,

$$\varnothing \Leftrightarrow \Delta, \ \cap \Leftrightarrow \cup, \ \subseteq \Leftrightarrow \supseteq .$$

Two given operators form a *dual pair* if their definitions are dual statements. Let f, g, g' be three set system operators such that g and g' form a dual pair. We say f and g are commutative if $f \circ g = g \circ f$, and call them anticommutative if

$$f \circ g = g' \circ f \text{ and } f \circ g' = g \circ f.$$

That is, traversing f through g will alter g to its dual operator g'.

2.1.3 Relevant Set Systems and Operators

An upper (or lower) system is a set system that contains all the supersets (or subsets) of its member structures. These two types of set systems are particularly important for our development due to their rich properties. Four operators are closely connected to upper and lower systems.

Definition 1. Given a set system Ω, we define the following four operators.

- $\uparrow \Omega := \{T \mid T \supseteq T' \text{ for some } T' \in \Omega\}$ (*Up-Closure Operator*),

- $\downarrow \Omega := \{T \mid T \subseteq T' \text{ for some } T' \in \Omega\}$ (*Down-Closure Operator*),
- $m(\Omega) := \{T \in \Omega \mid \forall T' \in \Omega, T' \subseteq T \implies T' = T\}$ (*Minimal Operator*),
- $M(\Omega) := \{T \in \Omega \mid \forall T' \in \Omega, T' \supseteq T \implies T' = T\}$ (*Maximal Operator*).

Clearly, these four operators form two dual pairs. We also have the following convenient properties associated with upper and lower systems. We omit their proofs since they can be derived directly.

Proposition 1. *Given a ground set Δ, upper systems in $\mathscr{P}^2(\Delta)$ have the following properties.*

- *$\varnothing$ and $\mathscr{P}(\Delta)$ are both upper systems,*
- *Ω is an upper system if and only if $\Omega = \uparrow \Omega$,*
- *Given a family of upper systems $\{\Omega_i\}_{i \in I}$, both $\bigcap_{i \in I} \Omega_i$ and $\bigcup_{i \in I} \Omega_i$ are upper systems,*
- *The complement of any upper system is a lower system,*
- *$\uparrow \Omega = \uparrow (m(\Omega))$,*
- *$m(\Omega) = m(\uparrow \Omega)$.*

The corresponding dual statements are also true for the lower systems.

The last two claims above indicate that the minimal elements in Ω capture all the information of the associated upper system. To develop the set system approximation theory, we further need the following operators.

Definition 2. Given a set system $\Omega \subseteq \mathscr{P}(\Delta)$, we define the following five operators.

- $\overline{\Omega} := \mathscr{P}(\Delta) \setminus \Omega$ (*Complement Operator*),
- $\widehat{\Omega} := \{\Delta \setminus T \mid T \in \Omega\}$ (*Element-Complement Operator*),
- $\mathcal{C}(\Omega) := \{S \mid \forall T \in \Omega,\ S \cap T \neq \varnothing\}$ (*Cut Operator*),
- $\mathcal{G}(\Omega) := \{S \mid \forall T \in \Omega,\ S \cup T \neq \Delta\}$ (*Cocut Operator*),
- $\mathcal{E}(\Omega) := \widehat{\mathcal{C}(\Omega)}$, i.e., the composition of $\mathcal{C}$ and $\widehat{(\cdot)}$ (*Elimination Operator*).

These operators have intuitive interpretations. Operator $\overline{(\cdot)}$ is designed to return all the non-solutions or non-structures within $\mathscr{P}(\Delta)$. Operator $\widehat{(\cdot)}$ is commonly applied in graph theory to generate the complement subgraph. For instance, suppose Ω is the set of independent sets in a network, then $\widehat{\Omega}$ contains all the vertex covers. We name $\mathcal{C}$ the cut operator because every element in $\mathcal{C}(\Omega)$ intersects every structure $T \in \Omega$, which mimics the relationship between s-t edge cuts and s-t paths in a given network. The significance of this operator in interdiction games has been studied in [37]. Clearly, the cut operator $\mathcal{C}$ and cocut operator $\mathcal{G}$ form a dual pair. The cocut operator can also be equivalently defined as follows,

$$\begin{aligned}\mathcal{G}(\Omega) &= \{S \mid \forall T \in \Omega,\ (\Delta \setminus S) \cap (\Delta \setminus T) \neq \varnothing\},\\ &= \{S \mid \forall T \in \Omega,\ (\Delta \setminus S) \nsubseteq T\}.\end{aligned}$$

The first says that the complement of S is a cut of the element-complement of Ω, and the second perspective implies that S ensures its complement does not conform to any structures in Ω. Finally, the elimination operator $\mathcal{E}$ is not as essential as the rest since all its properties can be derived from the cut and element-complement operators. We define this operator simply due to its connection with the well-known subtour elimination inequalities, which will be investigated later in Section 3.

Example 1. Consider the set system $\Omega := \Omega_1$ in Figure 1, we can compute the operators as follows.

- $\uparrow \Omega = \mathscr{P}(\Omega)$; $\downarrow \Omega = \downarrow \{\{1,3\},\{2,3\},\{1,2,4\}\}$;
- $m(\Omega) = \varnothing$; $M(\Omega) = \{\{1,3\},\{2,3\},\{1,2,4\}\}$;
- $\overline{\Omega} = \{\{3\},\{4\},\{1,2\},\{3,4\},\{1,2,3\},\{1,3,4\},\{2,3,4\},\Delta\}$;
- $\widehat{\Omega} = \{\{3\},\{1,3\},\{1,4\},\{2,3\},\{2,4\},\{1,3,4\},\{2,3,4\},\Delta\}$;
- $\mathcal{C}(\Omega) = \varnothing$; $\mathcal{G}(\Omega) = \downarrow \{\{1,2\},\{4\}\}$; $\mathcal{E}(\Omega) = \varnothing$.

The same computation can be carried out for the case $\Omega := \Omega_2$ in Figure 1. △

The above computations reveal some interesting properties. Notably, $\mathcal{C}$ and $\mathcal{E}$ lead to upper systems (more apparent in Ω_2) and $\mathcal{G}$ produces lower systems. These observations are part of a broader framework of calculation rules, designated as the *cut-cocut algebra*, which will be developed in the subsequent subsection and will be used throughout the paper thereafter.

2.2 Cut-Cocut Algebra

In this subsection, we aims to derive some basic algebraic properties for the previously defined operations. We need the following lemma.

Lemma 1. *For any dual operator pair g, g' defined on $\mathscr{P}^2(\Delta)$, we have*

$$\widehat{g(\cdot)} = g'\left(\widehat{(\cdot)}\right) \text{ and } \widehat{g'(\cdot)} = g\left(\widehat{(\cdot)}\right).$$

We organize the following theorem into five parts. The first four explore the basic properties associated with the first four operators in Definition 2, and the last part lists the corresponding interactions.

Theorem 1. *Given any $\Omega \subseteq \mathscr{P}(\Delta)$, we have the following properties,*

$\mathscr{A}$. *For the complement operator* $\overline{(\cdot)}$,

- $\mathscr{A}$*1.* $\overline{\varnothing} = \mathscr{P}(\Delta)$ *and* $\overline{\mathscr{P}(\Delta)} = \varnothing$
- $\mathscr{A}$*2.* Ω *and* $\overline{\Omega}$ *form a partition of* $\mathscr{P}(\Delta)$
- $\mathscr{A}$*3.* $\Omega \subseteq \Omega'$ *if and only if* $\overline{\Omega} \supseteq \overline{\Omega'}$ *(order-reversing)*
- $\mathscr{A}$*4.* Ω *is an upper (lower) system implies* $\overline{\Omega}$ *is a lower (upper) system*
- $\mathscr{A}$*5.* $\overline{\overline{\Omega}} = \Omega$ *(self-inverse)*

$\mathscr{B}$. *For the element-complement operator* $\widehat{(\cdot)}$,

- $\mathscr{B}$*1.* $\widehat{\varnothing} = \varnothing$ *and* $\widehat{\mathscr{P}(\Delta)} = \mathscr{P}(\Delta)$
- $\mathscr{B}$*2.* $\Omega \subseteq \Omega'$ *if and only if* $\widehat{\Omega} \subseteq \widehat{\Omega}'$ *(order-preserving)*
- $\mathscr{B}$*3.* $\widehat{(\cdot)}$ *acts as a contravariant functor on* Ω, *i.e., for every* $T, T' \in \Omega$ *such that* $T \subseteq T'$, *we have* $\Delta \setminus T, \Delta \setminus T' \in \widehat{\Omega}$ *with inclusion direction reversed*
- $\mathscr{B}$*4.* Ω *is an upper (lower) system implies* $\widehat{\Omega}$ *is a lower (upper) system*
- $\mathscr{B}$*5.* $\widehat{\widehat{\Omega}} = \Omega$ *(self-inverse)*

$\mathscr{C}$. *For the cut operator* $\mathcal{C}(\cdot)$,

- $\mathscr{C}$*1.* $\mathcal{C}(\varnothing) = \mathscr{P}(\Delta)$ *and* $\mathcal{C}(\mathscr{P}(\Delta)) = \varnothing$
- $\mathscr{C}$*2.* $\mathcal{C}(\Omega) = \varnothing$ *if and only if* $\varnothing \in \Omega$
- $\mathscr{C}$*3.* $\mathcal{C}(\Omega)$ *is an upper system*
- $\mathscr{C}$*4.* $\mathcal{C}(\Omega) = \mathcal{C}(m(\Omega))$
- $\mathscr{C}$*5.* $\Omega \subseteq \Omega'$ *implies* $\mathcal{C}(\Omega) \supseteq \mathcal{C}(\Omega')$ *(order-reversing)*
- $\mathscr{C}$*6.* $\mathcal{C}(\mathcal{C}(\Omega)) = \uparrow \Omega$ *(upper envelope)*

$\mathscr{D}$. *For the cocut operator* $\mathcal{G}(\cdot)$,

- $\mathscr{D}$*1.* $\mathcal{G}(\varnothing) = \mathscr{P}(\Delta)$ *and* $\mathcal{G}(\mathscr{P}(\Delta)) = \varnothing$
- $\mathscr{D}$*2.* $\mathcal{G}(\Omega) = \varnothing$ *if and only if* $\Delta \in \Omega$
- $\mathscr{D}$*3.* $\mathcal{G}(\Omega)$ *is a lower system*
- $\mathscr{D}$*4.* $\mathcal{G}(\Omega) = \mathcal{G}(M(\Omega))$
- $\mathscr{D}$*5.* $\Omega \subseteq \Omega'$ *implies* $\mathcal{G}(\Omega) \supseteq \mathcal{G}(\Omega')$ *(order-reversing)*
- $\mathscr{D}$*6.* $\mathcal{G}(\mathcal{G}(\Omega)) = \downarrow \Omega$ *(lower envelope)*

$\mathscr{E}$. *For interactions between operators,*

- $\mathscr{E}$*1.* $\widehat{\overline{\Omega}} = \overline{\widehat{\Omega}}$ *(commutative)*
- $\mathscr{E}$*2.* $\uparrow \widehat{\Omega} = \widehat{\downarrow \Omega}$ *and* $\downarrow \widehat{\Omega} = \widehat{\uparrow \Omega}$ *(anticommutative)*

ℰ3. $\widehat{M(\Omega)} = m\left(\widehat{\Omega}\right)$ *and* $\widehat{m(\Omega)} = M\left(\widehat{\Omega}\right)$ *(anticommutative)*

ℰ4. $\mathcal{C}(\widehat{\Omega}) = \widehat{\mathcal{G}(\Omega)}$ *and* $\mathcal{G}(\widehat{\Omega}) = \widehat{\mathcal{C}(\Omega)}$ *(anticommutative)*

ℰ5. $\overline{\uparrow\Omega} \subseteq \downarrow\overline{\Omega}$ *and* $\overline{\downarrow\Omega} \subseteq \uparrow\overline{\Omega}$ *(partially anticommutative)*

ℰ6. $m\left(\overline{\Omega}\right) \subseteq \overline{M\left(\Omega\right)}$ *and* $M\left(\overline{\Omega}\right) \subseteq \overline{m\left(\Omega\right)}$ *(partially anticommutative)*

ℰ7. $\mathcal{G}\left(\overline{\Omega}\right) \subseteq \overline{\mathcal{C}(\Omega)}$ *and* $\mathcal{C}\left(\overline{\Omega}\right) \subseteq \overline{\mathcal{G}(\Omega)}$ *with the equalities hold if and only if* Ω *is an upper system and a lower system, respectively. (partially anticommutative)*

Many other algebraic properties can be derived from the above theorem. We record some relevant ones below for later reference.

Corollary 1. *We have the following identities:*

- $\mathcal{G}(\cdot) = \mathcal{C}\left(\widehat{\widehat{(\cdot)}}\right)$;
- $\mathcal{E}(\cdot) = \widehat{\mathcal{C}(\cdot)} = \mathcal{G}\left(\widehat{(\cdot)}\right)$;
- $\mathcal{E}^2(\cdot) = \mathcal{G} \circ \mathcal{C}(\cdot)$;
- $\mathcal{G} \circ \mathcal{C}(\Omega) = \mathcal{P}(\Delta)$ *whenever* $\varnothing \in \Omega$*, and equals* $\varnothing$ *otherwise;*
- $\mathcal{C} \circ \mathcal{G}(\Omega) = \mathcal{P}(\Delta)$ *whenever* $\Delta \in \Omega$*, and equals* $\varnothing$ *otherwise.*

Hence, $\mathcal{G} \circ \mathcal{C}$ and $\mathcal{C} \circ \mathcal{G}$ can be considered as the empty set and ground set detectors for the input set system Ω. Utilizing all these established rules, we will develop the set system approximation method in the subsequent subsection.

2.3 Set System Approximation

Given a set system $\Omega \in \mathscr{P}^2(\Delta)$, our goal is to approximate it from inner and outer using simple systems, i.e., the upper and lower systems. We begin with the following definition and lemma.

Definition 3 (Embedding and Enclosing)**.** For two systems $\Omega \subseteq \Omega' \in \mathscr{P}^2(\Delta)$, we call Ω an embedding of Ω', and Ω' an enclosing of Ω.

Lemma 2 (Upper Embedding)**.** *For any set system* $\Omega \subseteq \mathscr{P}(\Delta)$*, we have*

$$\mathcal{C}\left(\widehat{\widehat{\Omega}}\right) \subseteq \Omega.$$

Moreover, this equality holds if and only if Ω *is an upper system.*

This theorem simply says that $\mathcal{C}\left(\widehat{\widehat{\Omega}}\right)$ characterizes a subset of the feasible solution space Ω. By ($\mathscr{C}3$), we know that $\mathcal{C}\left(\widehat{\widehat{\Omega}}\right)$ is also an upper system inside Ω. Moreover, when Ω itself is an upper system, this upper embedding is exact. Based on this, the following theorem provides the tightest inner and outer approximation for a given system.

Theorem 2 (Upper Approximation)**.** *In the same problem setting, we have*

$$\mathcal{C}\left(\widehat{\overline{\Omega}}\right) = \mathcal{C}\left(\widehat{M(\overline{\Omega})}\right) = \mathcal{C}\left(m\left(\widehat{\overline{\Omega}}\right)\right) \subseteq \Omega \subseteq \mathcal{C}^2(\Omega) = \uparrow \Omega = \mathcal{C}\left(\widehat{\overline{\uparrow \Omega}}\right),$$

where $\mathcal{C}\left(\widehat{\overline{\Omega}}\right)$ *and* $\mathcal{C}\left(\widehat{\overline{\uparrow \Omega}}\right)$ *are the tightest upper embedding and enclosing of* Ω*, respectively. Moreover, the equality holds for all if and only if* Ω *is an upper set.*

This theorem says that the two upper set systems $\mathcal{C}\left(\widehat{\overline{\Omega}}\right)$ and $\mathcal{C}\left(\widehat{\overline{\uparrow \Omega}}\right)$ sandwiches Ω in the tightest fashion, which can provide the best inner and outer approximations using upper set systems. Symmetrically, it is expected to have the following dual statement for lower approximation. We omit the proof as it is simply the dual statement of Theorem 2.

Corollary 2 (Lower Approximation)**.** *Given a set system* Ω*, we have*

$$\mathcal{G}\left(\widehat{\overline{\Omega}}\right) = \mathcal{G}\left(\widehat{m\left(\overline{\Omega}\right)}\right) = \mathcal{G}\left(M\left(\widehat{\overline{\Omega}}\right)\right) \subseteq \Omega \subseteq \mathcal{G}^2(\Omega) = \downarrow \Omega = \mathcal{G}\left(\widehat{\overline{\downarrow \Omega}}\right),$$

where $\mathcal{G}\left(\widehat{\overline{\Omega}}\right)$ *and* $\mathcal{G}\left(\widehat{\overline{\downarrow \Omega}}\right)$ *are the tightest lower embedding and enclosing of* Ω*, respectively. Moreover, the equality holds for all if and only if* Ω *is a lower system.*

These results provide the tightest inner and outer approximation methods for an arbitrary set system Ω using upper and lower systems. They can be further extended to other interesting cases by the following corollary.

Corollary 3. *Given a set system* Ω*, the tightest upper embeddings with respect to* $\overline{\Omega}$*,* $\widehat{\Omega}$*, and* $\widehat{\overline{\Omega}}$ *are*

$$\mathcal{C}(\widehat{\Omega}) \subseteq \overline{\Omega},\ \mathcal{C}(\overline{\Omega}) \subseteq \widehat{\Omega},\ \mathcal{C}(\Omega) \subseteq \widehat{\overline{\Omega}},$$

and their equalities hold if and only if Ω *is a lower system for the first two cases and is an upper set for the third case. Symmetrically,* $\mathcal{G}(\widehat{\Omega})$*,* $\mathcal{G}(\overline{\Omega})$*, and* $\mathcal{G}(\Omega)$ *are the respective tightest lower embeddings with equality conditions reversed.*

To ease future notation, we define the upper and lower approximation operators as follows.

Definition 4 (Approximation Operators)**.** Given a set system Ω, we define

$$\tilde{\mathcal{C}}(\cdot) := \mathcal{C}\left(\widehat{\overline{(\cdot)}}\right)$$
$$\tilde{\mathcal{G}}(\cdot) := \mathcal{G}\left(\widehat{\overline{(\cdot)}}\right)$$

as the upper/lower approximation operators.

Example 2. Consider the set system $\Omega := \Omega_1$ in Figure 1. We can directly compute $\tilde{\mathcal{C}}(\Omega)$ and $\tilde{\mathcal{C}}(\uparrow \Omega)$ to obtain the trivial set systems $\varnothing$ and $\mathscr{P}(\Delta)$, and derive the following systems as

$$\tilde{\mathcal{G}}(\Omega) = \{\varnothing, \{1\}, \{2\}\},\ \tilde{\mathcal{G}}(\downarrow \Omega) = \downarrow \{\{1,3\}, \{2,3\}, \{1,2,4\}\}.$$

It is easy to verify that these are indeed the tightest inner and outer approximations using upper and lower systems. Similarly, for the case $\Omega := \Omega_2$, we have the tightest upper embedding and enclosing as

$$\tilde{\mathcal{C}}(\Omega) = \uparrow \{\{1,2\},\{1,4\}\} \subseteq \Omega_2 \subseteq \tilde{\mathcal{C}}(\uparrow \Omega) = \uparrow \{\{1\}\},$$

while the tightest lower embedding and enclosing are the trivial set systems. △

3 Approximation Reformulations for BIPs

In this section, we will establish a reformulation framework for the general BIP (2) utilizing the set system approximation results. We will also develop general constraint separation subroutines for these reformulations and analyze their time complexity. Throughout this section, we consider $\Omega := \Omega_{\mathcal{X}}$ as the set system corresponding to the feasible region $\mathcal{X}$ of Problem (2).

3.1 Approximation Reformulations

The inner and outer approximation methods developed in Section 2 have two advantageous properties: (i) they are the tightest inner and outer approximations using upper and lower systems, which has been shown in the last section; (ii) they are described by the cut and cocut operators that are closely related to the set covering and subtour elimination inequalities by the following lemma.

Lemma 3. *Given any set system Ω, the vector representations of $\mathcal{C}(\Omega)$ and $\mathcal{E}(\Omega) = \mathcal{G}(\widehat{\Omega})$ are equivalent to the following* covering *and* elimination *inequalities, respectively.*

$$\mathcal{X}_{\mathcal{C}(\Omega)} = \left\{ x \in \{0,1\}^n \;\middle|\; \sum_{i \in T} x_i \geq 1, \forall T \in \Omega \right\},$$
$$\mathcal{X}_{\mathcal{E}(\Omega)} = \left\{ x \in \{0,1\}^n \;\middle|\; \sum_{i \in T} x_i \leq |T| - 1, \forall T \in \Omega \right\}.$$

Moreover, these two types of inequalities can be equivalently converted to each other by the substitution $y := 1 - x$.

This lemma together with the set system approximation results developed from the last section enable the following approximation reformulation methods. We use $z(\Pi) \in \mathbb{R} \cup \{\pm\infty\}$ to denote the optimal value of the optimization problem Π, where $+\infty$ and $-\infty$ indicate the infeasible and unbounded scenarios.

Theorem 3. *Any general BIP (2) can be reformulated into the following forms for inner/outer approximations,* *Upper Embedding* Π_{ui}:

$$\min_{x\in\{0,1\}^n} f(x) \tag{3a}$$
$$s.t. \sum_{i\in T} x_i \geq 1,\ \forall T \in m\left(\widehat{\overline{\Omega}}\right) \tag{3b}$$

Upper Enclosing Π_{uo}:

$$\min_{x\in\{0,1\}^n} f(x) \tag{4a}$$
$$s.t. \sum_{i\in T} x_i \geq 1,\ \forall T \in m\left(\widehat{\overline{\uparrow\Omega}}\right) \tag{4b}$$

Lower Embedding Π_{li}:

$$\min_{x\in\{0,1\}^n} f(x) \tag{5a}$$
$$s.t. \sum_{i\in T} x_i \leq |T|-1,\ \forall T \in m\left(\overline{\Omega}\right) \tag{5b}$$

Lower Enclosing Π_{lo}:

$$\min_{x\in\{0,1\}^n} f(x) \tag{6a}$$
$$s.t. \sum_{i\in T} x_i \leq |T|-1,\ \forall T \in m\left(\overline{\downarrow\Omega}\right) \tag{6b}$$

with objective values satisfying

$$z(\Pi_{uo}) \leq z(\Pi) \leq z(\Pi_{ui}), \quad z(\Pi_{lo}) \leq z(\Pi) \leq z(\Pi_{li}).$$

Moreover, (3) *and* (4) *are both equivalent to* (2) *if and only if* Ω *is an upper system;* (5) *and* (6) *are both equivalent to* (2) *if and only if* Ω *is a lower system.*

Many properties of these reformulations can be derived from the cut-cocut algebra in Theorem 1. We list below the relevant ones regarding the feasibility and redundancy conditions of the four approximation reformulations.

Corollary 4. *The feasibility and redundancy conditions of the approximations are the following*

- Π_{ui} *is infeasible iff* $\Delta \notin \Omega$, *and is redundant iff* $\Omega = \mathscr{P}(\Delta)$;
- Π_{uo} *is infeasible iff* $\Omega = \varnothing$, *and is redundant iff* $\varnothing \in \Omega$;
- Π_{li} *is infeasible iff* $\varnothing \notin \Omega$, *and is redundant iff* $\Omega = \mathscr{P}(\Delta)$;
- Π_{lo} *is infeasible if and only if* $\Omega = \varnothing$, *and is redundant if and only if* $\Delta \in \Omega$.

In practical applications, the covering and elimination constraint sets in Lemma 3 usually consist of a considerable number of inequalities. This requires the use of an iterative cut generation approach for their separation. In the following subsection, we will introduce general separation subroutines and conduct an analysis of the associated time complexity.

3.2 A General Separation Procedure

For optimal computational performance of the reformulations outlined in Theorem 3, efficient separation subroutines are essential. While these subroutines can be specially designed for various instance problems by leveraging their unique solution space Ω, this subsection focuses on the development of general (integer) separation procedures applicable whenever Ω is an upper or lower

Algorithm 1 Separation Subroutine for (3b) with an Upper Set Ω

input: a ground set Δ, a membership oracle $\mathbb{I}_\Omega$, current solution T_x
output: a structure $T \in \widehat{\overline{\Omega}}$ or $\varnothing$
if $\mathbb{I}_\Omega(T_x) = 1$ **then return** $\varnothing$
$C \leftarrow$ any inclusion chain from T_x to Δ
$S \leftarrow \texttt{BinarySearch}(C, \mathbb{I}_\Omega)$
return $\Delta \setminus S$

Algorithm 2 Binary Search for (3b) with an Upper Set Ω

input: an inclusion chain C, a membership oracle $\mathbb{I}_\Omega$
output: the maximum structure $S \in C \cap \overline{\Omega}$
if $|C| = 1$ **then return** the only element in the chain C
$k \leftarrow \texttt{Int}(|C|/2)$
if $\mathbb{I}_\Omega(C[k]) = 1$ **then**
  $C \leftarrow C[: k-1]$
else
  $C \leftarrow C[k :]$
end if
return $\texttt{BinarySearch}(C, \mathbb{I}_\Omega)$

system. We will also prove that this separation subroutine is poly-time solvable whenever an efficient membership oracle of Ω is accessible. In the next section, we will extend this method to address arbitrary set systems.

Clearly, a lower system in (3) or an upper system in (5) will lead to a redundant or infeasible solution space by Corollary 4. Moreover, when Ω is either upper or lower, the other two reformulations (4) and (6) are equivalent to (3) and (5), respectively. Therefore, we can only focus on two cases: separating (3b) with an upper system Ω and separating (5b) with a lower system Ω. We will show that, with a membership oracle for Ω, both types of separation procedures can be implemented efficiently. We need the following definition.

Definition 5 (Inclusion Chain)**.** Given $T \subseteq T' \subseteq \Delta$, the *inclusion chain* C from T to T' is a sequence of structures $C := (T_0, T_1, \ldots, T_k)$ where $T_0 = T$, $T_k = T'$, and each pair of consecutive structures increases only by exactly one element.

For (3b) with an upper system Ω, Algorithms 1 and 2 together provide a general separation procedure given a membership oracle for Ω, where the BinarySearch subroutine conducts a binary search on an inclusion chain. The following theorem proves the correctness and provides the time complexity.

Theorem 4. *Given an upper system $\Omega \neq \varnothing$ as the solution space of* (3)*, a structure $T_x \subseteq \Delta$, and a chain C from T_x to Δ. Algorithm 1 returns $\varnothing$ whenever T_x is feasible, and returns a structure in $\widehat{\overline{\Omega}}$ associated with the maximum element in $C \cap \overline{\Omega}$ if otherwise. Let $O(\tau(\Omega))$ be the complexity of the membership oracle $\mathbb{I}_\Omega$, the complexity for separating such an inequality in* (3b) *is* $O(\log(|\Delta| - |T_x| + 1) \cdot \tau(\Omega))$.

Algorithm 3 Separation Subroutine for (5b) with a Lower Set Ω

input: a ground set Δ, a membership oracle $\mathbb{I}_\Omega$, current solution T_x
output: a structure $T \in \overline{\Omega}$ or $\varnothing$
if $\mathbb{I}_\Omega(T_x) = 1$ **then return** $\varnothing$
$C \leftarrow$ any inclusion chain from $\varnothing$ to T_x
return `BinarySearch`$(C, \mathbb{I}_\Omega)$

Algorithm 4 Binary Search for (5b) with a Lower Set Ω

input: an inclusion chain C, a membership oracle $\mathbb{I}_\Omega$
output: the minimum structure $S \in C \cap \overline{\Omega}$
if $|C| = 1$ **then return** the only element in the chain C
$k \leftarrow$ `Int`$(|C|/2)$
if $\mathbb{I}_\Omega(C[k]) = 1$ **then**
  $C \leftarrow C[k+1:]$
else
  $C \leftarrow C[:k]$
end if
return `BinarySearch`$(C, \mathbb{I}_\Omega)$

Similarly, we provide the separation subroutine for (5b) in Algorithm 3 and 4 along with the following theorem for correctness and complexity.

Theorem 5. *Given a lower system $\Omega \neq \varnothing$ as the solution space of (5), a structure $T_x \subseteq \Delta$, and a chain C from $\varnothing$ to T_x. Algorithm 3 returns $\varnothing$ whenever T_x is feasible, and returns a minimum structure in $C \cap \overline{\Omega}$ if otherwise. Let $O(\tau(\Omega))$ be the complexity of the membership oracle $\mathbb{I}_\Omega$, the complexity for separating such an inequality in (5b) is $O(\log(|T_x|+1) \cdot \tau(\Omega))$.*

We omit the proof since it is almost identical to the previous one with only two differences: (i) the index set for (5b) is $\overline{\Omega}$; (ii) Ω is a lower system. In most optimization models, the membership oracle can be implemented as evaluating a given x on the corresponding constraint set. Thus, it is often efficient. Therefore, both separation subroutines are generally efficient.

3.3 Facets Analysis

One advantage of the reformulations in (3)–(6) is their ability to leverage existing facet analysis results for a range of BIPs. This subsection presents general results under the assumption that Ω is a monotone set, starting with the following lemma and definition.

Lemma 4. *Given any binary solution space $\mathcal{X} := \{x \in \{0,1\}^n \mid g(x) \leq 0\}$, define*

$$\widehat{\mathcal{X}} := \{x \in \{0,1\}^n \mid g(1-x) \leq 0\}.$$

Then, the function $\theta(x) = 1 - x$ establishes a bijective affine transformation from $\operatorname{conv}(\mathcal{X})$ to $\operatorname{conv}(\widehat{\mathcal{X}})$. In particular, an inequality $\langle a, x\rangle \geq b$ is facet-defining for $\operatorname{conv}(\mathcal{X})$ if and only if $\langle a, \theta(x)\rangle \geq b$ is facet-defining for $\operatorname{conv}(\widehat{\mathcal{X}})$.

Definition 6 (Quasi-Feasibility)**.** A non-solution $T \in \overline{\Omega}$ is called quasi-feasible if for every $a \in T$ there exists some $a' \in \Delta \setminus T$ such that $T \setminus \{a\} \cup \{a'\} \in \Omega$; it is called co-quasi-feasible if for every $a' \in \Delta \setminus T$ there exists some $a \in T$ such that $T \setminus \{a\} \cup \{a'\} \in \Omega$.

With this lemma and definition, we can reinterpret the facet characterization for set covering polytope in [3, 36] for the general BIPs.

Theorem 6. *Suppose Ω is an upper set and $|T| \geq 2$ for every $T \in m(\widehat{\overline{\Omega}})$, the covering inequality (3b) is facet-defining if and only if the associated maximal non-solution $\Delta \setminus T \in M(\overline{\Omega})$ is quasi-feasible. Symmetrically, suppose Ω is a lower set and $|T| \geq 2$ for every $T \in m(\overline{\Omega})$, the elimination inequality (5b) is facet-defining if and only if the associated minimal non-solution $T \in m(\overline{\Omega})$ is co-quasi-feasible.*

This theorem illustrates how the proposed reformulation framework can connect the extensive literature on set covering problems to the analysis of various BIPs and their solution subspaces.

3.4 Examples

In this subsection, we present three examples to demonstrate the application of the proposed set system approximation framework. It is important to note that these examples are not meant to suggest that the proposed reformulations are the most suitable solution approach for these specific problems. Rather, our focus is on showcasing the structures that emerge from applying the proposed method and exploring the insights they offer for some classic BIPs.

Since all examples in this subsection are related to graphs, we will introduce the basic notation set. Given a graph G, $V(G)$ and $E(G)$ are the corresponding vertex and edge sets, which will be denoted simply by V and E when the underlying graph G is clear from the context. Given any vertex and edge subsets $V' \subseteq V$ and $E' \subseteq E$, we use $G(V')$ and $G(E')$ for their induced subgraphs. For notation simplicity, we do not differ the vertex/edge set and their induced subgraphs whenever such distinction is inessential. For any $v \in V$, we use $\deg(v)$ to denote the degree of v, i.e., the number of edges incident with v. Finally, we define $\mathrm{cl}(G)$ as the closure of G obtained by repeatedly adding a new edge $e = (u, v) \in E$ for nonadjacent vertices u and v with $\deg(u) + \deg(v) \geq n$ until no more such edges can be found.

Example 3 (s-t Paths and s-t Edge Cuts)**.** Consider the following shortest path problem

$$\Pi_{\mathrm{SP}}: \min_{x \in \mathcal{X} \subseteq \{0,1\}^n} \langle c, x \rangle,$$

where $\Omega := \Omega_{\mathcal{X}}$ contains all the s-t paths and c is a non-negative vector that represents the edge lengths. Since the objective function is increasing on x, we can extend the solutions space Ω to contain all s-t paths along with their supersets (in terms of edges). Hence, we have

$$\Omega := \{s\text{-}t \text{ connected subgraphs}\},$$

which is clearly an upper system. Then, we compute as follows

- $\overline{\Omega} :=$ {subgraphs that does not connect s and t};
- $\widehat{\overline{\Omega}} :=$ {s-t cuts and their supersets}.

By Theorem 2, we have $\mathcal{C}\left(\widehat{\overline{\Omega}}\right) = \Omega$. That is, T is feasible (i.e., a subgraph that connects s and t) if and only if it interdicts all s-t cuts.

From the other direction, suppose $\mathcal{X}$ represents the space of all the s-t cuts. Then, the above problem, denoted by Π_{MC}, becomes the minimum s-t cut problem. By the same argument, we can see that T is feasible (an s-t cut) if and only if it interdicts all s-t paths. This echoes a certain dual relationship between s-t cuts and s-t connected subgraphs. △

Example 4 (Simple s-t Paths and s-t Cocuts)**.** Consider the longest simple path problem

$$\Pi_{\text{LP}}: \min_{x \in \mathcal{X} \subseteq \{0,1\}^n} \langle -c, x \rangle,$$

where $\Omega_{\mathcal{X}}$ contains all the simple s-t paths and c is a non-negative vector represents the edge lengths. To conform to the minimization convention, we compute the negative path length instead. Since the objective function is decreasing on x, we can extend the solutions space to contain all simple paths and their subsets (in terms of edge sets) as follows.

$$\Omega := \{\text{edge sets that can be extended to some } s\text{-}t \text{ path}\},$$

and reformulate the problem using (5), which gives,

$$\begin{aligned} \Pi_{\text{LP}}: \min_{x \in \{0,1\}^n} \ & \langle -c, x \rangle \\ \text{s.t.} \ & \sum_{i \in T} x_i \leq |T| - 1, \ \forall T \in m(\overline{\Omega}). \end{aligned}$$

By definition, we have,

- $\overline{\Omega} :=$ {edge sets that cannot be extended to an s-t path};
- $\widehat{\overline{\Omega}} :=$ {edge sets that cannot form a complete graph with any s-t path}.

We call the latter s-t cocuts. To separate elements from $\overline{\Omega}$, we can directly generate claws (star with three edges) and cycles since they are subgraphs that cannot extended to any s-t paths, which rediscovers the findings in [22] from a new perspective. Besides these two cases, any other edge set $T \in \overline{\Omega}$ must be a family of vertex-disjoint subpaths $\{p_i\}_{i \in I}$. To test whether T can be extended to a s-t path, we note that a valid s-t path extended from T must connect every subpath p_i through its two endpoints v_i, u_i without using any edges in p_i. Thus, we can define $U = \bigcup_{i \in I}\{v_i, u_i\} \cup \{s, t\}$ and find all the pairwise paths between vertices in U without using any edges in p_i (remove all these edges when computing the paiwise paths). Then, a depth-first-search from s to t using these obtained pairwise paths provides a valid verifier. Since Ω is a lower system, generating all elimination inequalities for these cocuts is an exact solution approach for this longest simple path problem. This provides a new cut generation method for solving the longest path problem with solely the elimination type of inequalities, which is quite different from other proposed formulations in the literature [27]. $\triangle$

From the two examples discussed above, we observe that when s-t paths are the primary structures under consideration, the resulting cuts and cocuts lead to markedly distinct structures. This reflects the fundamental disparity between the classic shortest and longest path problems. We conduct a similar analysis below for TSP as follows.

Example 5 (Hamiltonian Cycles)**.** Let Ω be the set of Hamiltonian cycles (HCs), then

$$\uparrow\Omega = \mathcal{C}\left(\widehat{\overline{\uparrow\Omega}}\right) \text{ and } \downarrow\Omega = \mathcal{G}\left(\widehat{\overline{\downarrow\Omega}}\right)$$

are the tightest enclosings of Ω using upper and lower systems. Similar to the shortest and longest path problems, when the objective function is monotone, we can identify Ω with its upper or lower closure without affecting the optimization result. In particular, we can use covering inequalities indexed by $\widehat{\overline{\uparrow\Omega}}$ to describe $\uparrow\Omega = \{$edge sets that contain any HC$\}$ and use elimination constraints labeled by $\overline{\downarrow\Omega}$ to characterize $\downarrow\Omega = \{$edge sets that are contained by any HC$\}$. It is well-known that checking whether a general graph contains an HC is NP-compete, thus testing the membership of $\overline{\uparrow\Omega}$, i.e., checking whether a graph does not contain any HC, falls into the co-NP-complete category. However, the following proposition provides some special types of subgraphs that do not contain any HC, which can be used to produce constraints in (4).

Proposition 2. *Given $G = (V, E)$ with $|V| \geq 3$, $T \in \overline{\uparrow\Omega}$ if T satisfies any of the following conditions,*

- *$G(T)$ is disconnected;*
- *for some $v \in V(G(T))$, $deg(v) = 1$;*
- *there exists a vertex bipartition such that a non-singleton part has at most one vertex connected to the other part;*
- *$G(T) = cl(G(T'))$ for any above T'.*

Note that the third condition generalizes the first two. This proposition provides a sufficient yet not necessary condition, thus the corresponding covering constraints are valid but not sufficient to ensure an optimal solution. In contrast, we have the following exact characterization for $\overline{\downarrow\Omega}$.

Proposition 3. *Given $G = (V, E)$, $T \in \overline{\downarrow\Omega}$ if and only if T satisfies any of the following conditions,*

- *T contains a none-Hamiltonian cycle (NHC);*

- *T contains a claw (i.e., a star with three edges);*
- *T is a set of mutually disjoint paths that cannot be extended to any HC.*

Moreover, when G is a complete graph, the first two types constitute the entire $\overline{\downarrow \Omega}$.

Through the set system approximation lens, it is somewhat counterintuitive that traditional formulations for TSP (i.e., shortest HC problem) prefer elimination inequalities over covering inequalities. Indeed, given that the objective function increases with x, Ω could be equated to its upper closure $\uparrow \Omega$, mirroring our approach in the shortest path example. This would make covering inequalities appear more naturally for derivation. However, Proposition 2 highlights the challenge in defining the structures within $\overline{\uparrow \Omega}$. As a result, classic TSP formulations focus on describing Ω by excluding all non-maximal structures from $\downarrow \Omega$. In essence, they employ elimination inequalities for non-Hamiltonian cycles (NHCs). Then, the additional constraints that ensure each vertex has a degree of two are used for two purposes: (i) eliminate the second and third types of structures, and (ii) remove every strict subset of any HC from the solution space. Note that the NHC elimination along with (i) exactly characterizes $\downarrow \Omega$ by Proposition 3, and (ii) removes those non-maximal structures in $\downarrow \Omega$.

From this perspective, the classic TSP formulations are indeed unique. Their distinctiveness does not stem from the use of elimination inequalities, as these are commonly employed in characterizing lower systems. Rather, it is the integration of specific constraints to exclude non-maximal structures from the lower closure that sets them apart. △

4 Method Extension

Reformulations in Theorem 3 are exact if and only if the solution space Ω is an upper or lower system. Otherwise, they are only approximation methods, which could be quite loose in many cases. For instance, when Ω does not contain $\varnothing$ or Δ, there are no upper nor lower embeddings, which renders both (3) and (5) useless. In this section, we develop two orthogonal approaches to extend the cases where these reformulations are still exact: (i) formulation standardization techniques so that Ω can be augmented to an upper or lower system without affecting the optimization problem; (ii) set system decomposition method to describe an arbitrary Ω as the union and intersection of a set of upper and lower systems.

4.1 Formulation Standardization

We will introduce two standardization techniques that could transform the original problem representation (2) to a more convenient form for applying the reformulation methods in Theorem 3. The main idea is to tailor the original set system Ω to an upper (or lower) set without affecting the optimal solutions.

Intuitively, when using the upper embedding for approximating minimization problems, we aim to tighten the lower part of the set system Ω so that optimal solutions are near the minimal elements at the bottom, and widen the upper part to contain a larger upper system for a better embedding. For instance, consider $\Omega := \Omega_2$ in Figure 1, suppose we know $\{1\}$ is not an optimal solution to the associated objective function, we can remove it from Ω so that the minimal elements become $\{1,2\}$ and $\{1,4\}$, and the resulting set system becomes upward closed. This allows an exact characterization using reformulations in Theorem 3. Similarly, when using lower approximation, we will prefer the opposite, i.e., tightening the upper and widening the lower. To achieve this, we propose the following standardization steps.

Definition 7 (Optimality-Invariant Standardization I)**.** Given Problem (2), we perform two steps to standardize the solution space $\mathcal{X}$ before applying the approximation reformulations.

i. Tightening: remove identifiable non-optimal solutions in $\mathcal{X}$ to obtain $\mathcal{X}'$.

ii. Widening: for upper (lower) approximation, add all the supersets (subsets) of $T \in \Omega_{\mathcal{X}'}$ that does not affect the optimality of the problem to obtain $\mathcal{X}''$.

We call the resulting solution space $\mathcal{X}''$ an optimality-invariant counterpart of $\mathcal{X}$.

The benefit of this standardization procedure can be illustrated by an unrealistic extreme example. Suppose we can tighten $\mathcal{X}$ to a singleton to represent a specific structure T, and then widen $\Omega_{\mathcal{X}}$ to contain all the supersets of T without affecting the optimality. Then, a direct computation shows that

$$\tilde{\mathcal{C}}(\Omega) = \{\{i\}\}_{i \in T}.$$

Then, both (3b) and (4b) will produce the strongest inequalities

$$x_i \geq 1, \ \forall i \in T$$

to obtain the optimal solution that represents T.

In particular, for monotone objective functions, the solution space can be consistently expanded to include all supersets or subsets without altering the optimal solutions. We present this result below without proof, as its validity is self-evident.

Corollary 5. *Suppose f is increasing (decreasing) on $\mathcal{X}$, then $\uparrow \Omega$ ($\downarrow \Omega$) is an optimality-invariant counterpart of Ω. In particular, the upper (lower) approximations are exact in this case.*

For instance, linear or polynomial functions with nonnegative (or non-positive) coefficients are monotone on $\mathscr{P}(\Delta)$. However, being monotone for general functions f is still quite a strong restriction. To extend the scope, we define the following class of functions that can be easily transformed into a monotone function.

Definition 8 (Entry-wise Monotone Functions)**.** Given $f : \{0,1\}^n \to \mathbb{R}$, let (k, x_{-i}) denote a vector from $\{0,1\}^n$ where the ith entry is equal to $k \in \{0,1\}$ and the rest entries are equal to $x_{-i} \in \{0,1\}^{n-1}$, and define

$$\Delta_{x_{-i}} = f(1, x_{-i}) - f(0, x_{-i}).$$

Then, we say f is entry-wise monotone if for every $i \in [n]$, the sign of $\Delta_{x_{-i}}$ will not change by changing $x_{-i} \in \{0,1\}^{n-1}$.

This definition includes a much larger class of functions. We list some common functions in the following proposition.

Proposition 4. *The following functions are entry-wise monotone:*

- *Monotone functions.*
- *Linear (modular) functions.*
- *For every $i \in \Delta$, $\partial_i f([0,1]^n)$ is entirely contained in $\mathbb{R}_+$ or $\mathbb{R}_-$.*
- *Submodular functions f where for every $i \in \Delta$, either*

$$f(\{i\}) - f(\varnothing) \leq 0 \text{ or } f(\Delta) - f(\Delta \setminus \{i\}) \geq 0.$$

- *Subadditive functions f where $f(e_i) \leq 0$ for all $i \in \Delta$.*
- *Supermodular and superadditive counterparts.*

The following proposition provides a standardization method for this class of functions.

Proposition 5 (Optimality-Invariant Standardization II)**.** *For Problem (2) with an entry-wise monotone objective function, we can obtain an equivalent reformulation with an increasing (or decreasing) function by setting $x_i = 1 - x'_i$ for all $i \in I := \{i \in \Delta \mid \Delta_{x_{-i}} < 0\}$ (or $I := \{i \in \Delta \mid \Delta_{x_{-i}} > 0\}$).*

With both standardization methods, we are able to transform a larger class of problems (2) to have upper or lower systems as solution spaces, so that the reformulations in Theorem 3 become exact. For cases where such standardization methods do not apply, we introduce a set system decomposition methodology in the next subsection.

4.2 Set System Decomposition

We aim to decompose any given set system Ω as the combination of unions and intersections of upper/lower systems. We begin with the following definition and lemma.

Definition 9 (Interval Set System)**.** A set system Ω is said to have the interval property if for every $T, T' \in \Omega$, any T'' sandwiched between T and T' is also in Ω.

Lemma 5. *A set system Ω has the interval property if and only if it can be written as $\Omega = \Omega_1 \cap \Omega_2$ for some upper system Ω_1 and some lower system Ω_2.*

This lemma indicates that the following reformulation is exact for interval systems.

Corollary 6. *Suppose the solution space* Ω *of Problem* (2) *is an interval set system, then the following is an exact reformulation of* (2)*.*

$$\min_{x\in\{0,1\}^n} \ f(x) \tag{7a}$$
$$s.t. \ \sum_{i\in T} x_i \geq 1, \qquad \forall T \in m\left(\widehat{\uparrow\Omega}\right), \tag{7b}$$
$$\sum_{i\in T} x_i \leq |T| - 1, \ \forall T \in m\left(\overline{\downarrow\Omega}\right). \tag{7c}$$

Moreover, Algorithm 1 and 3 can be used to separate constraints in (7b) *and* (7c) *with membership oracles for* $\uparrow\Omega$ *and* $\downarrow\Omega$*.*

Using the nice properties of interval set systems, we aim to decompose a general set system Ω into the union of a finite number of interval systems. This is indeed achievable according to the following theorem.

Theorem 7 (Set System Decomposition Theorem)**.** *Every set system* Ω *can be decomposed as* $\Omega = \bigcup_{k\in K}\Omega_k$ *for a finite family of interval set systems* $\{\Omega_k\}_{k\in K}$*.*

Proof. We provide a rather explicit decomposition as follows,

$$\Omega = \bigcup_{T\in\Omega} \left(\uparrow\{T\}\cap\downarrow\{T\}\right).$$

Clearly, each singleton $\uparrow\{T\}\cap\downarrow\{T\} = \{T\}$ is a trivial interval set. □

The decomposition outlined in this theorem serves primarily for theoretical validation, as its practical application is hindered by the often excessively large size of $|\Omega|$. In practical scenarios, the goal is to identify a decomposition where the cardinality $|K|$ is minimized. Integrating these observations leads us to the subsequent corollary. We omit the proof since it is an immediate consequence of previous results.

Corollary 7. *Given any set system* Ω *with a decomposition* $\Omega = \bigcup_{k\in K}\Omega_k$ *for some interval set systems* $\{\Omega_k\}_{k\in K}$*, problem* (2) *can be equivalently reformulated as*

$$\min_{x\in\{0,1\}^n, z\in\{0,1\}^{|K|}} \ f(x) \tag{8a}$$
$$s.t. \ \sum_{i\in T} x_i \geq z_k, \qquad \forall k\in K, \forall T \in m\left(\widehat{\uparrow\Omega_k}\right), \tag{8b}$$
$$\sum_{i\in S} x_i \leq |S| - z_k, \ \forall k\in K, \forall S \in m\left(\overline{\downarrow\Omega_k}\right), \tag{8c}$$
$$\sum_{k\in K} z_k = 1. \tag{8d}$$

Moreover, Algorithm 1 and 3 can be used to separate constraints in (7b) *and* (7c) *with membership oracles for each* $\uparrow\Omega_k$ *and* $\downarrow\Omega_k$*.*

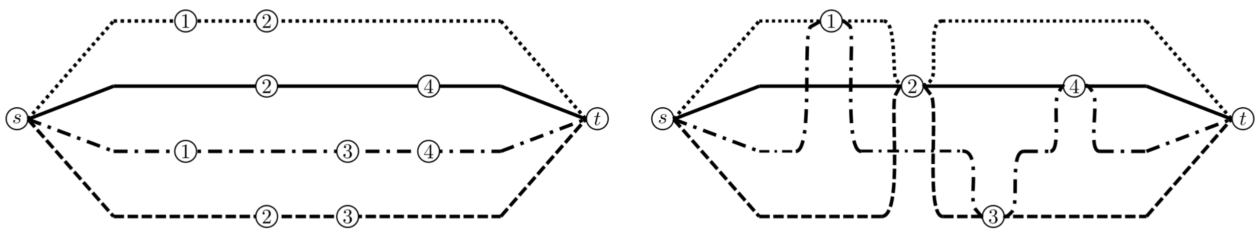

Figure 2: General Max-Flow Min-Cut Theorem. Given $m(\Omega) = \{\{1,2\},\{2,3\},\{2,4\},\{1,3,4\}\}$, we first construct one pipe for each $T \in m(\Omega)$, and connects these pipes with two artificial nodes s and t (left figure). Then, we merge nodes on different pipes and assign c_i as the capacities on the nodes with infinite capacity on the rest of the pipes (right figure). Pipes can share capacity at each node, but flows do not cross pipes.

Interestingly, this reformulation can be split into K copies of (7) and solved in parallel. Moreover, the best incumbent solution among them can be shared across all copies to improve the overall efficiency.

5 Structural Duality

In Example 3, we observe the close relationship between s-t connected subgraphs and s-t edge cuts. This mimics the classic flow-cut duality since the underlying graph structures for the max-flow and min-cut problems are indeed the s-t connected subgraphs and s-t edge cuts. Interestingly, the derivation of the duality in Example 3 relies on the usage of set operators instead of linear programming duality theory. In this section, we utilize this tool to develop a structural duality theory for arbitrary set systems. We begin with the following definition.

Definition 10 (Structural Duality)**.** Given any upper (lower) set system $\Omega \subseteq \Delta$, we call $\widehat{\widehat{\Omega}}$ the upper (lower) dual system of Ω. We also call the pair of dual structures $\left(\Omega, \widehat{\widehat{\Omega}}\right)$ a *cut* (*cocut*) pair. Moreover, for an arbitrary set system Ω, we call $\widehat{\widehat{\uparrow \Omega}}$ and $\widehat{\widehat{\downarrow \Omega}}$ the cut and cocut completions, respectively.

The following proposition provides some basic properties of a dual structural pair, which can be directly derived from previous results.

Proposition 6. *A cut (cocut) pair* $\left(\Omega, \widehat{\widehat{\Omega}}\right)$ *satisfies the following properties:*

- *Every pair of* $T \in \Omega$ *and* $T' \in \widehat{\widehat{\Omega}}$ *cuts (cocuts) each other, i.e.,* $T \cap T' \neq \varnothing$ $(T \cup T' \neq \Delta)$*.*
- $\mathcal{C}(\Omega) = \widehat{\widehat{\Omega}}$ *and* $\mathcal{C}\left(\widehat{\widehat{\Omega}}\right) = \Omega$ *for every cut pair, and* $\mathcal{G}(\Omega) = \widehat{\widehat{\Omega}}$ *and* $\mathcal{G}\left(\widehat{\widehat{\Omega}}\right) = \Omega$ *for every cocut pair.*

In the next theorem, we show that this type of structural dual pair is indeed a generalization of the classic min-cut max-flow duality.

Theorem 8 (General Min-Cut Max-Flow Duality)**.** *For any cut pair* Ω *and* $\widehat{\widehat{\Omega}}$ *with a non-negative cost vector* c *assign to the ground set* Δ*, there exists a flow function* f *with* c *as the capacity vector such that*

$$\min_{x \in \mathcal{X}_{\widehat{\Omega}}} \langle c, x \rangle = \max_{x \in \mathcal{X}_{\Omega}} f(c, x), \tag{9}$$

and f *is increasing on both* c *and* x*.*

The intuition of this proof is illustrated in Figure 2. We note that this max-flow problem is impractical to implement as it requires knowing all the minimal structures in $m(\Omega)$ to construct the graph. Instead, this theorem mainly serves as a conceptual vehicle to generalize the structural aspect of classic max-flow min-cut theorem to all the binary optimization problems whenever $\Omega_{\mathcal{X}}$ is an upper or lower system. This further confirms that the structural dual pair defined in Definition 10 is indeed a generalization of the classic duality relationship between the s-t connected subgraphs and s-t cuts.

5.1 Dual Pairs of Common Structures

In this subsection, we explore the dual pairs associated with commonly encountered set systems and graph structures.

Example 6 (Spanning Trees)**.** Given a graph $G = (V, E)$, a spanning tree is a connected subgraph without cycles. Let Ω be the set of spanning trees, then $\uparrow \Omega$ and $\downarrow \Omega$ are the set of connected subgraphs and the set of forests, respectively. A direct computation shows that

$$\widehat{\uparrow \Omega} = \{T \subseteq E \mid G(E \setminus T) \text{ is disconnected}\},$$
$$\widehat{\downarrow \Omega} = \{T \subseteq E \mid G(\Delta \setminus T) \text{ contains a cycle}\}.$$

The former and the latter are called the edge cuts and the co-cycles, respectively. Hence, connected subgraphs and edge cuts form a cut pair, and the forests and co-cycles form a cocut pair. △

Example 7 (Knapsack (Set Packing) Problem)**.** The classic knapsack problem aims to identify an item list that is below the knapsack's capacity w to optimize the given objective function. Hence, the solution space Ω_w consists of all item lists that are under the given capacity w, which is clearly a lower system. Let W be the total item weight and $\epsilon > 0$ be a sufficiently small value, a direct computation shows that the corresponding dual structure is the following,

$$\widehat{\widehat{\Omega}} = \Omega_{w'} \text{ with } w' = W - w - \epsilon.$$

Hence, Ω_w and $\Omega_{w'}$ form a cocut pair. △

Example 8 (Vertex Covers & Independent Sets)**.** Given a graph $G = (V, E)$, a vertex cover is a vertex subset $T \subseteq V$ that includes at least one endpoint of every edge of the graph. The system Ω

of vertex covers is clearly an upper system. Then, a direct computation shows the dual structures is the following,

$$\widehat{\widehat{\Omega}} = \uparrow \{\{u,v\} \in E\}.$$

That is, the vertex sets that contain at least both endpoints of an edge. Since independent sets form the lower system $\widehat{\Omega}$, its cocut counterpart is equal to

$$\overline{\widehat{\widehat{\Omega}}} = \overline{\Omega} = \downarrow \{V \setminus \{u,v\} \mid \{u,v\} \in E\}.$$

This is, the vertex sets that exclude both endpoints of at least one edge. △

Example 9 (Matchings)**.** Given a graph $G = (V,E)$, a feasible matching is a set of edges $T \subseteq E$ that have mutually disjoint vertices. Hence, the system of matchings Ω is lower closed. Then, a direct computation shows that its cocut counterpart is the following.

$$\overline{\widehat{\Omega}} = \downarrow \{T \subseteq E \mid G(E \setminus T) \text{ is a connected subgraph with exactly two edges}\}.$$

Hence, one way to solve maximum matching is to apply elimination inequalities on $\overline{\Omega}$, i.e., the set of connected subgraphs with exactly two edges. △

All these structural dual pairs reveal certain interesting relations between different types of set systems and graph structures. In addition, they can directly lead to the reformulations introduced in Theorem 3.

6 Case Study: Network Site Selection under Uncertainty

To illustrate the versatility of the proposed reformulation framework, we apply it to a case study on the network site selection problem under uncertainty.

6.1 Problem Setting

Consider a supply network $G = (V,E)$ where a company aims to select certain nodes from V to supply products to all demand nodes. If constructed, each site $i \in V$ would induce a fixed net benefit r_i, which incorporates both economic and environmental factors, and thus can be either positive or negative. The potential supplies from all nodes to node j is denoted by the vector $a_j = (a_{ij})_{i\in V} \in \mathbb{R}_+^{|V|}$, and the demand at node j is $b_j \in \mathbb{R}_+$. Historical data indicates that, for every $j \in V$, (a_j, b_j) is a random vector following some empirical joint distribution $\bar{\mathbb{P}}_j$ with the support denoted by Ξ_j. To avoid service overlap, the company also requires that the selected sites $T \subseteq V$ form an independent set. Then, the following formulation uses chance constraints to provide demand satisfaction guarantee.

$$\max_{x\in\{0,1\}^V} \sum_{i\in V} r_i x_i \tag{10a}$$

$$\text{s.t. } \bar{\mathbb{P}}_j \left(\sum_{i \in \delta[j]} a_{ij} x_i \geq b_j \right) \geq 1 - \epsilon_j, \ \forall j \in V \tag{10b}$$

$$\sum_{i \in C} x_i \leq 1, \qquad \forall C = \{i, j\} \in E. \tag{10c}$$

The objective (10a) is to maximize the net benefit. (10b) contains the chance constraints to impose demand satisfaction requirements with $\delta[j]$ denoting the neighboring vertices of j including j itself, and (10c) ensures that the resulting solution is an independent set. We can further strengthen (10c) by expanding the index set E to the set of cliques in G.

Since $(a_j, b_j) \geq 0$ for all $j \in V$, the binary spaces $\mathcal{X}_1, \mathcal{X}_2$ of x described by (10b) and (10c) correspond to an upper system and a lower system, respectively. Thus, the entire solution space $\mathcal{X}_1 \cap \mathcal{X}_2$ is an interval system. Note that (10c) is already in the elimination form (7c). Then, by a direct computation, $\hat{\overline{\Omega}}_{\mathcal{X}_1}$ contains every vertex set that is the complement of some unsatisfied site selection. Given any fixed solution x and the nominal distribution $\bar{\mathbb{P}}_j$, the probability in (10b) can either be evaluated directly or approximated using finite scenario sampling [35].

6.2 Performance Comparison of Four Implementations

Since the proposed method supports cut generation for arbitrary binary spaces or subspaces, it enables flexible algorithmic choices. We compare the following four implementations.

- NoCut: finite scenario approximation (FSA) for (10b) with exact constraints in (10c);
- ClqCut: FSA for (10b) with maximal clique cut generation for (10c);
- SatCut: cut generation for (10b) with exact constraints in (10c);
- AllCut: cut generation for both (10b) and (10c).

In NoCut and SatCut implementations, we rewrite (10b) into the following forms

$$\sum_{i \in \delta[j]} a_{ij}^k x_i + b_j^k (1 - z_j^k) \geq b_j^k, \qquad \forall k \in [K], j \in V$$

$$\sum_{k \in [K]} z_j^k / K \geq 1 - \epsilon_j, \ \forall j \in V,$$

where a_{ij}^k's and b_j^k's are sampled from $\bar{\mathbb{P}}_j$. In ClqCut, we generate constraints in (10c) on-the-fly by identifying at most three non-vertex cliques in the vertex-induced subgraph $G[T]$ for some x_T from the master problem. In SatCut, given such a x_T, the jth subproblem simply generates K samples of (a_j, b_j) from $\bar{\mathbb{P}}$ to approximate the satisfaction probability. This will be used to either confirm or reject the feasibility of x_T regarding (10b). For the rejection case, the proposed constraint

$$\sum_{i \in V \setminus T} x_i \geq 1 \tag{11}$$

Config	Runtime				Cuts Sep. Time			Num. of Cuts		
(n, m, ϵ, K)	NoCut	ClqCut	SatCut	AllCut	ClqCut	SatCut	AllCut	ClqCut	SatCut	AllCut
$(40, 234, 0.05, 200)$	1.54	1.66	**0.10**	0.15	**0.00**	0.04	0.08	**32.00**	32.3	152.0
$(40, 234, 0.05, 500)$	3.72	3.11	**0.08**	0.15	**0.01**	0.03	0.07	21.0	**11.67**	32.0
$(40, 234, 0.05, 800)$	6.43	6.43	**0.23**	0.43	**0.01**	0.09	0.27	28.0	**19.00**	130.7
$(40, 234, 0.1, 200)$	1.30	1.20	**0.06**	0.08	**0.00**	0.01	0.04	19.0	**18.33**	66.7
$(40, 234, 0.1, 500)$	3.52	3.13	**0.14**	0.25	**0.01**	0.05	0.14	20.0	**18.67**	105.3
$(40, 234, 0.1, 800)$	6.36	6.16	**0.16**	0.32	**0.01**	0.06	0.16	20.0	**12.67**	76.0
$(40, 546, 0.05, 200)$	3.13	2.41	**0.04**	0.04	**0.00**	0.02	0.03	17.0	**8.33**	28.0
$(40, 546, 0.05, 500)$	7.24	7.13	**0.06**	0.10	**0.01**	0.02	0.07	17.0	**2.33**	28.0
$(40, 546, 0.05, 800)$	12.03	10.13	**0.13**	0.16	**0.01**	0.07	0.11	17.0	**8.67**	28.0
$(40, 546, 0.1, 200)$	2.90	2.54	**0.04**	0.06	**0.00**	0.01	0.04	20.0	**5.00**	34.7
$(40, 546, 0.1, 500)$	8.32	6.24	**0.07**	0.83	**0.02**	**0.02**	0.07	15.0	**6.00**	28.0
$(40, 546, 0.1, 800)$	12.72	11.32	**0.15**	0.24	**0.03**	0.08	0.16	25.0	**12.67**	42.7
$(80, 948, 0.05, 200)$	5.95	6.29	**0.18**	0.26	**0.03**	0.04	0.15	43.0	**15.00**	122.7
$(80, 948, 0.05, 500)$	15.26	13.68	0.96	**0.25**	**0.03**	0.05	0.15	21.0	**12.67**	33.3
$(80, 948, 0.05, 800)$	40.36	33.25	**0.46**	0.94	**0.03**	0.16	0.55	50.0	**25.33**	168.0
$(80, 948, 0.1, 200)$	6.46	6.60	**0.31**	0.34	**0.01**	0.08	0.20	51.0	**44.00**	258.7
$(80, 948, 0.1, 500)$	18.36	14.31	**0.38**	0.54	**0.02**	0.10	0.34	32.0	**21.67**	116.0
$(80, 948, 0.1, 800)$	41.80	32.37	**0.53**	0.67	**0.02**	0.14	0.38	32.0	**13.00**	61.3
$(80, 2212, 0.05, 200)$	12.69	11.10	**0.13**	0.19	0.05	**0.03**	0.13	27.0	**9.67**	41.3
$(80, 2212, 0.05, 500)$	37.65	31.67	**0.24**	0.46	**0.08**	**0.08**	0.32	34.0	**12.00**	57.3
$(80, 2212, 0.05, 800)$	61.83	48.16	**0.24**	0.37	**0.07**	**0.07**	0.27	20.0	**5.33**	28.0
$(80, 2212, 0.1, 200)$	12.69	11.24	**0.16**	0.24	0.05	**0.03**	0.15	43.0	**12.67**	64.0
$(80, 2212, 0.1, 500)$	40.00	36.09	**0.25**	0.48	0.12	**0.06**	0.31	42.0	**11.33**	69.3
$(80, 2212, 0.1, 800)$	62.77	49.44	**0.29**	1.29	0.25	**0.08**	0.33	21.0	**8.00**	28.0
$(120, 2142, 0.05, 200)$	16.79	13.60	1.74	**0.64**	**0.03**	0.07	0.22	50.0	**21.00**	109.3
$(120, 2142, 0.05, 500)$	71.53	42.37	**1.18**	1.55	**0.03**	0.20	0.89	51.0	**34.33**	288.0
$(120, 2142, 0.05, 800)$	94.18	65.81	**0.84**	0.85	**0.05**	0.13	0.41	48.0	**13.67**	49.3
$(120, 2142, 0.1, 200)$	15.41	13.20	0.78	**0.69**	**0.03**	0.06	0.20	46.0	**19.67**	128.0
$(120, 2142, 0.1, 500)$	76.46	39.29	**1.05**	1.12	**0.03**	0.15	0.48	47.0	**16.33**	98.7
$(120, 2142, 0.1, 800)$	105.53	66.04	**1.20**	1.72	**0.04**	0.24	1.02	47.0	**18.67**	182.7
$(120, 4998, 0.05, 200)$	39.27	36.82	**0.49**	2.74	0.57	**0.02**	0.79	38.0	**6.00**	34.7
$(120, 4998, 0.05, 500)$	114.57	97.42	**0.76**	1.57	0.65	**0.02**	1.01	40.0	**9.00**	78.7
$(120, 4998, 0.05, 800)$	180.60	146.16	**1.29**	2.51	2.03	**0.02**	1.31	41.0	**4.00**	45.3
$(120, 4998, 0.1, 200)$	37.72	30.48	**0.21**	0.93	0.62	**0.02**	0.66	20.0	**8.33**	29.3
$(120, 4998, 0.1, 500)$	120.46	90.71	**0.56**	1.79	0.51	**0.02**	0.84	36.0	**5.00**	34.7
$(120, 4998, 0.1, 800)$	190.05	150.03	**0.73**	1.71	1.30	**0.02**	0.90	38.0	**9.00**	40.0

Table 1: The comparison of four algorithms for the site selection problem with satisfaction constraints. Three performance metrics are considered runtime, cuts separation time, and the number of generated cuts. For each instance configuration, the minimum value in each performance category is highlighted. Overall, SatCut significantly outperforms the other algorithms, with AllCut closely following.

will be added to the master according to (7b).

The experiment was conducted on a 2023 MacBook Pro with an M2 Max chip featuring 12 CPU cores and 64 GB of memory, using Python 3.9 as the programming language and Gurobi 10.0.1 as the optimization solver. We assume each a_{ij} is supported on $[0, 1]$, following a Beta distribution Beta$(2, 2)$ if $\{i, j\} \in E$, and $a_{ij} = 0$ otherwise. The values d_i are uniformly distributed over $[0, 0.1]$. The benefit coefficients r_i are integers randomly generated between -20 and 20.

The parameters used in the experiment are $n \in \{40, 80, 120\}$, $m \in \{0.3n(n-1)/2, 0.7n(n-1)/2\}$, $\epsilon \in \{0.05, 0.1\}$, and $K \in \{200, 500, 800\}$, representing the number of vertices, number of edges, demand violation tolerance, and number of sampling scenarios, respectively. Each tuple (n, m, ϵ, K) is referred to as an instance configuration, with the two choices of m corresponding to graph instances with density 0.3 and 0.7, respectively. For each configuration, we generate three connected Erdős–Rényi graphs, resulting in a total of 108 instances. The four algorithms are then executed on these instances with 600 seconds time limit for optimization. The corresponding results are summarized in Table 1.

In terms of average runtime, SatCut and AllCut significantly outperform the other two implementations, with SatCut having a slight edge in efficiency. Implementing cut generation for (10c) using maximal clique cuts improves runtime performance compared to the NoCut algorithm, though not to the same extent as the other two methods. From the "Num. of Cuts" column, we observe that SatCut achieves this performance with the fewest cuts, indicating that these are strong inequalities effectively tightening the upper bound. Additionally, larger vertex sizes, higher graph density, and more scenarios generally require more computational time, a trend that is more pronounced in the NoCut and ClqCut algorithms. In contrast, the violation tolerance ϵ has a minimal impact on computational complexity.

Overall, the proposed reformulation framework offers diverse solution strategies for analysis and comparison. In our instances, generating cuts for (10b) with fixed constraints in (10c) demonstrates the best performance. We also note that the $|V|$ subproblems for generating (11) can be further parallelized due to the constraint-wise independence in (10b), which could further improve the efficiency of the SatCut and AllCut implementations.

6.3 Distributionally Robust Chance Constraints

In practice, the nominal distribution $\bar{\mathbb{P}}_j$ is often deviated from the true distribution. To hedge against such ambiguity and improve the out-of-sample performance, the following distributionally robust version of (10b) is often used.

$$\sup_{\mathbb{P}_j \in \mathfrak{P}(\bar{\mathbb{P}}_j)} \mathbb{P}_j \left(\sum_{i \in \delta[j]} a_{ij} x_i < b_j \right) = \sup_{\mathbb{P}_j \in \mathfrak{P}(\bar{\mathbb{P}}_j)} \mathbb{E}_{\mathbb{P}_j} \left[\mathbb{I}_{\Xi_j(x)} \right] \leq \epsilon_j, \ \forall j \in V, \tag{12}$$

where $\Xi_j(x) := \{(a_j, b_j) \in \Xi \mid \sum_{i \in \delta[j]} a_{ij} x_i < b_j\}$, $\mathbb{I}_\Xi$ is the set indicator function of Ξ, and $\mathfrak{P}(\bar{\mathbb{P}}_j)$ is some ambiguity set around the nominal distribution $\bar{\mathbb{P}}_j$. It should be clear that binary solution space $\mathcal{X}_1$ restricted to (12) is still an upper set. Thus, the same type of cut as in (11) can be generated whenever the above inequality is violated.

We use the standard Wasserstein type-1 ball $W_1(\mathbb{P}_j, \bar{\mathbb{P}}_j) := \inf_{\pi \in \Pi(\mathbb{P}_j, \bar{\mathbb{P}}_j)} \mathbb{E}_\pi[\|(a_j, b_j) - (a'_j, b'_j)\|]$ to construct the following ambiguity set $\mathfrak{P}(\bar{\mathbb{P}}_j) := \{\mathbb{P} \mid W_1(\mathbb{P}, \bar{\mathbb{P}}_j) \leq \eta\}$, where $\| \cdot \|$ is chosen to be the 2-norm and $\eta > 0$ is the associated radius. When the nominal distribution $\bar{\mathbb{P}}_j$ is supported on finite scenarios $\{(a_j^k, b_j^k)\}_{k\in[K]}$, we have the following duality result according to [28].

$$\sup_{\mathbb{P}_j \in \mathfrak{P}(\bar{\mathbb{P}}_j)} \mathbb{E}_{\mathbb{P}_j}\left[\mathbb{I}_{\Xi_j(x)}\right] = \inf_{s_j^k, \gamma_j} \sum_{k\in[K]} s_j^k/K + \eta\gamma_j$$
$$\text{s.t. } s_j^k + \gamma_j \|(a_j, b_j) - (a_j^k, b_j^k)\| \geq \mathbb{I}_{\Xi_j(x)}(a_j, b_j), \quad \forall k \in [K], (a_j, b_j) \in \Xi_j.$$

Note that the set indicator function is not piecewise concave, thus we cannot use the reformulation method introduced in [28]. Instead, we can use the following finite scenario approximation method assuming Ξ is fully supported on the samples.

$$\begin{aligned} \inf_{s_j^k, \gamma_j} \ & \sum_{k\in[K]} s_j^k/K + \eta\gamma_j \\ \text{s.t. } & s_j^k + \gamma_j \|(a_j^l, b_j^l) - (a_j^k, b_j^k)\| \geq \mathbb{I}_{\Xi_j(x)}(a_j^l, b_j^l), \quad \forall k, l \in [K]. \end{aligned}$$

In the DRO-SatCut implementation, we use this linear programming as the subproblem for each $j \in V$ to separate (11). For comparison, the DRO-NoCut algorithm incorporates the above formulation as constraints to obtain the following MIP,

$$\begin{aligned} \max_{x\in\{0,1\}^V, s_j^k, \gamma_j} \ & \sum_{i\in V} r_i x_i \\ \text{s.t. } & \sum_{k\in[K]} s_j^k/K + \eta\gamma_j \leq \epsilon_j, && \forall j \in V \\ & s_j^k + \gamma_j \|(a_j^l, b_j^l) - (a_j^k, b_j^k)\| \geq 1 - z_j^l, && \forall k, l \in [K] \\ & \sum_{i\in\delta[j]} a_{ij}^k x_i + b_j^k (1 - z_j^k) \geq b_j^k, && \forall k \in [K], j \in V \\ & \sum_{i\in C} x_i \leq 1, && \forall C = \{i, j\} \in E. \end{aligned}$$

Using the same experiment setting as before with the radius $\eta = 0.1$, we present the experiment results in Table 2.

According to this table, DRO-SatCut consistently outperforms DRO-NoCut across all configurations. While DRO-NoCut frequently hits time limits in configurations with 500 and 800 sampling scenarios, DRO-SatCut maintains solution times within seconds. This efficiency is achieved by spending the majority of execution time on separating the set covering constraints (11). This significant performance enhancement highlights the effectiveness of the proposed reformulation and cut generation framework. Similar as before, the $|V|$ subproblems in DRO-SatCut can be parallelized to further enhance time efficiency.

Config	DRO-NoCut		DRO-SatCut		
(n, m, ϵ, K)	Runtime	Gap	Runtime	Cust Sep. Time	Num. of Cuts
$(40, 234, 0.05, 200)$	48.46	0.00	**0.52**	0.45	32.3
$(40, 234, 0.05, 500)$	204.08	0.00	**0.81**	0.76	11.7
$(40, 234, 0.05, 800)$	–	0.48	**4.40**	4.27	19.0
$(40, 234, 0.1, 200)$	25.08	0.00	**0.22**	0.18	18.3
$(40, 234, 0.1, 500)$	222.69	0.00	**1.24**	1.14	18.7
$(40, 234, 0.1, 800)$	–	0.14	**2.75**	2.64	12.7
$(40, 546, 0.05, 200)$	28.33	0.00	**0.21**	0.19	8.3
$(40, 546, 0.05, 500)$	173.33	0.00	**0.34**	0.30	2.3
$(40, 546, 0.05, 800)$	357.70	0.00	**2.73**	2.67	8.7
$(40, 546, 0.1, 200)$	25.07	0.00	**0.11**	0.09	5.0
$(40, 546, 0.1, 500)$	197.59	0.00	**0.44**	0.40	6.0
$(40, 546, 0.1, 800)$	497.01	0.00	**2.87**	2.80	12.7
$(80, 948, 0.05, 200)$	168.69	0.00	**0.50**	0.37	15.0
$(80, 948, 0.05, 500)$	468.79	0.00	**0.95**	0.85	12.7
$(80, 948, 0.05, 800)$	–	–	**6.79**	6.47	25.3
$(80, 948, 0.1, 200)$	197.36	0.00	**1.04**	0.81	44.0
$(80, 948, 0.1, 500)$	–	0.10	**2.49**	2.22	21.7
$(80, 948, 0.1, 800)$	–	–	**5.03**	4.62	13.0
$(80, 2212, 0.05, 200)$	119.53	0.00	**0.30**	0.20	9.7
$(80, 2212, 0.05, 500)$	–	0.09	**2.20**	2.04	12.0
$(80, 2212, 0.05, 800)$	–	–	**1.34**	1.17	5.3
$(80, 2212, 0.1, 200)$	103.32	0.00	**0.36**	0.25	12.7
$(80, 2212, 0.1, 500)$	–	0.21	**1.96**	1.76	11.3
$(80, 2212, 0.1, 800)$	–	–	**3.38**	3.17	8.0
$(120, 2142, 0.05, 200)$	365.42	0.00	**1.43**	0.60	21.0
$(120, 2142, 0.05, 500)$	–	0.78	**4.84**	3.88	34.3
$(120, 2142, 0.05, 800)$	–	–	**4.14**	3.40	13.7
$(120, 2142, 0.1, 200)$	325.72	0.00	**1.27**	0.50	19.7
$(120, 2142, 0.1, 500)$	–	0.76	**3.14**	2.25	16.3
$(120, 2142, 0.1, 800)$	–	–	**7.60**	5.91	18.7
$(120, 4998, 0.05, 200)$	191.21	0.00	**1.42**	0.26	6.0
$(120, 4998, 0.05, 500)$	–	–	**1.94**	1.24	9.0
$(120, 4998, 0.05, 800)$	–	–	**2.97**	2.61	4.0
$(120, 4998, 0.1, 200)$	152.23	0.00	**0.46**	0.28	8.3
$(120, 4998, 0.1, 500)$	–	0.46	**1.20**	0.72	5.0
$(120, 4998, 0.1, 800)$	–	–	**3.20**	2.56	9.0

Table 2: The comparison of two algorithms for the site selection problem with DRO satisfaction constraints. A dash in the Runtime and Gap columns indicates that the algorithm exceeded the time limit and did not obtain the optimality gap, respectively. For each instance configuration, the minimum runtime is highlighted. Overall, DRO-SatCut significantly outperforms DRO-NoCut across all instances.

7 Conclusion

This paper presents a generic reformulation technique capable of transforming any given BIP into a format involving only covering and elimination inequalities, along with a singular choice constraint. This wide applicability is achieved through the development of set system approximation theory, grounded in the principles of cut-cocut algebra. This fresh analytical perspective not only facilitates a new reformulation technique but also connects advancements in set covering and elimination inequalities to arbitrary BIPs. For instance, Theorem 6 interprets facet characterization in set covering polytope to general BIPs.

This entire framework offers a flexible and convenient toolkit for problem analysis and solution strategy development. As demonstrated in the case study in Section 6, applying the reformulation method to different parts of the constraints can result in varied solution strategies and efficiency improvements. The approximation reformulations from Theorem 3 also provide a mechanism to identify inequalities for internal or external approximation, as illustrated in the examples in Section 3.4. Additionally, the developed general constraint separation subroutine and facet-defining conditions present a unified framework for implementing and analyzing various BIPs.

These developments also open up avenues for future research. One area of interest is whether set operators corresponding to other types of inequalities, such as packing and partition constraints, exist, and if so, how their algebraic interactions with other operators might enhance BIP reformulations. Another interesting direction is the potential discovery of efficient set system decomposition algorithms for specific system types. Overall, by investigating the algebraic properties of set systems and their operators, our framework introduces a new analytical toolkit for BIPs, potentially sparking further research and enriching the field.

References

[1] N. Achuthan, L. Caccetta, and S. Hill. A new subtour elimination constraint for the vehicle routing problem. *European Journal of Operational Research*, 91(3):573–586, 1996.

[2] A. Atamtürk. Cover and pack inequalities for (mixed) integer programming. *Annals of Operations Research*, 139(1):21–38, 2005.

[3] E. Balas and S. M. Ng. On the set covering polytope: I. all the facets with coefficients in {0, 1, 2}. *Mathematical Programming*, 43(1):57–69, 1989.

[4] E. Balas and M. W. Padberg. On the set-covering problem. *Operations Research*, 20(6):1152–1161, 1972.

[5] M. Bellmore and G. L. Nemhauser. The traveling salesman problem: a survey. *Operations Research*, 16(3):538–558, 1968.

[6] J. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numer. Math*, 4(1):238–252, 1962.

[7] K. E. Booth, T. T. Tran, and J. C. Beck. Logic-based decomposition methods for the travelling purchaser problem. In *Integration of AI and OR Techniques in Constraint Programming: 13th International Conference, CPAIOR 2016, Banff, AB, Canada, May 29-June 1, 2016, Proceedings 13*, pages 55–64. Springer, 2016.

[8] Q. Botton, B. Fortz, L. Gouveia, and M. Poss. Benders decomposition for the hop-constrained survivable network design problem. *INFORMS journal on computing*, 25(1):13–26, 2013.

[9] D. S. Chisca, M. Lombardi, M. Milano, and B. O'Sullivan. Logic-based benders decomposition for super solutions: An application to the kidney exchange problem. In *Principles and Practice of Constraint Programming: 25th International Conference, CP 2019, Stamford, CT, USA, September 30–October 4, 2019, Proceedings 25*, pages 108–125. Springer, 2019.

[10] R. Church and C. ReVelle. The maximal covering location problem. In *Papers of the regional science association*, volume 32, pages 101–118. Springer-Verlag Berlin/Heidelberg, 1974.

[11] A. R. Clark, R. M. Neto, and E. A. Toso. Multi-period production setup-sequencing and lot-sizing through atsp subtour elimination and patching. In *Proceedings of the 25th workshop of the UK planning and scheduling special interest group. University of Nottingham*, pages 80–87, 2006.

[12] G. Codato and M. Fischetti. Combinatorial benders' cuts for mixed-integer linear programming. *Operations Research*, 54(4):756–766, 2006.

[13] M. Desrochers and G. Laporte. Improvements and extensions to the miller-tucker-zemlin subtour elimination constraints. *Operations Research Letters*, 10(1):27–36, 1991.

[14] C. Gao, N. Wei, and J. L. Walteros. An exact approach for solving pickup-and-delivery traveling salesman problems with neighborhoods. *Transportation Science*, 57(6):1560–1580, 2023.

[15] Z. Gu, G. L. Nemhauser, and M. W. Savelsbergh. Lifted cover inequalities for 0-1 integer programs: Computation. *INFORMS Journal on Computing*, 10(4):427–437, 1998.

[16] O. Günlük. A branch-and-cut algorithm for capacitated network design problems. *Mathematical Programming*, 86:17–39, 1999.

[17] J. Hooker. *Logic-based benders decomposition: theory and applications*. Springer Nature, 2023.

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

[19] E. Israeli and R. K. Wood. Shortest-path network interdiction. *Networks: An International Journal*, 40(2):97–111, 2002.

[20] K. Kaparis and A. N. Letchford. Local and global lifted cover inequalities for the 0–1 multidimensional knapsack problem. *European journal of operational research*, 186(1):91–103, 2008.

[21] I. Kara, G. Laporte, and T. Bektas. A note on the lifted miller–tucker–zemlin subtour elimination constraints for the capacitated vehicle routing problem. *European Journal of Operational Research*, 158(3):793–795, 2004.

[22] W. Kern and D. Paulusma. Contracting to a longest path in h-free graphs. *arXiv preprint arXiv:1810.01542*, 2018.

[23] G. Laporte, Y. Nobert, and D. Arpin. An exact algorithm for solving a capacitated location-routing problem. *Annals of Operations Research*, 6:291–310, 1986.

[24] C. Lee, D. Cho, and S. Park. A combinatorial benders decomposition algorithm for the directed multiflow network diversion problem. *Military Operations Research*, 24(1):23–40, 2019.

[25] Y. Li, J.-F. Côté, L. Callegari-Coelho, and P. Wu. Novel formulations and logic-based benders decomposition for the integrated parallel machine scheduling and location problem. *INFORMS journal on computing*, 34(2):1048–1069, 2022.

[26] L. Lozano and J. C. Smith. A backward sampling framework for interdiction problems with fortification. *INFORMS Journal on Computing*, 29(1):123–139, 2017.

[27] R. G. Marzo, R. A. Melo, C. C. Ribeiro, and M. C. Santos. New formulations and branch-and-cut procedures for the longest induced path problem. *Computers & Operations Research*, 139:105627, 2022.

[28] P. Mohajerin Esfahani and D. Kuhn. Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1):115–166, 2018.

[29] I. B. Mohamed, W. Klibi, R. Sadykov, H. Şen, and F. Vanderbeck. The two-echelon stochastic multi-period capacitated location-routing problem. *European journal of operational research*, 306(2):645–667, 2023.

[30] Y. Park, P. Nielsen, and I. Moon. Unmanned aerial vehicle set covering problem considering fixed-radius coverage constraint. *Computers & Operations Research*, 119:104936, 2020.

[31] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei. The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3):801–817, 2017.

[32] M. Sánchez-García, M. I. Sobrón, and B. Vitoriano. On the set covering polytope: Facets with coefficients in $\{0, 1, 2, 3\}$. *Annals of Operations Research*, 81(0):343–356, 1998.

[33] A. Sassano. On the facial structure of the set covering polytope. *Mathematical Programming*, 44(1-3):181–202, 1989.

[34] T. H. Smith, V. Srinivasan, and G. Thompson. Computational performance of three subtour elimination algorithms for solving asymmetric traveling salesman problems. In *Annals of Discrete Mathematics*, volume 1, pages 495–506. Elsevier, 1977.

[35] Y. Song, J. R. Luedtke, and S. Küçükyavuz. Chance-constrained binary packing problems. *INFORMS Journal on Computing*, 26(4):735–747, 2014.

[36] N. Wei and J. L. Walteros. Integer programming methods for solving binary interdiction games. *European Journal of Operational Research*, 302(2):456–469, 2022.

[37] N. Wei and J. L. Walteros. On supervalid inequalities for binary interdiction games. *Mathematical Programming*, pages 1–42, 2024.

[38] N. Wei, J. L. Walteros, and F. M. Pajouh. Integer programming formulations for minimum spanning tree interdiction. *INFORMS Journal on Computing*, 33(4):1461–1480, 2021.

[39] Y. Yao, T. Edmunds, D. Papageorgiou, and R. Alvarez. Trilevel optimization in power network defense. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(4):712–718, 2007.

[40] Q. Yu, Y. Adulyasak, L.-M. Rousseau, N. Zhu, and S. Ma. Team orienteering with time-varying profit. *INFORMS Journal on Computing*, 34(1):262–280, 2022.

[41] E. Zemel. Easily computable facets of the knapsack polytope. *Mathematics of Operations Research*, 14(4):760–764, 1989.

A Mathematical Proofs

Lemma 1. *For any dual operator pair g, g' defined on $\mathscr{P}^2(\Delta)$, we have*

$$\widehat{g(\cdot)} = g'\left(\widehat{(\cdot)}\right) \text{ and } \widehat{g'(\cdot)} = g\left(\widehat{(\cdot)}\right).$$

Proof. Note that $\widehat{(\cdot)}$ acts as a contravariant functor on $\mathscr{P}(\Delta)$, i.e., for every $T, T' \in \Omega$ such that $T \subseteq T'$, we have $\Delta \setminus T, \Delta \setminus T' \in \widehat{\Omega}$ with inclusion direction reversed. In particular, $\widehat{\mathscr{P}(\Delta)}$ can be considered the same as $\mathscr{P}(\Delta)$ with all the inclusion direction reversed. Let $\theta : \mathscr{P}(\Delta) \to \widehat{\mathscr{P}(\Delta)}$ be such a functor that maps every object A to $\Delta \setminus A$ and every arrow $A \subseteq B$ to $\Delta \setminus A \supseteq \Delta \setminus B$, then any logical statement

$$p(S, \varnothing, \Delta, A_1 \cap B_1, A_2 \cup B_2, A_3 \subseteq B_3, A_4 \supseteq B_4)$$

is true in $\mathscr{P}(\Delta)$ if and only if its contravariant statement

$$p(\theta(S), \Delta, \varnothing, \theta(A_1) \cup \theta(B_1), \theta(A_2) \cap \theta(B_2), \theta(A_3) \supseteq \theta(B_3), \theta(A_4) \subseteq \theta(B4))$$

is true in $\widehat{\mathscr{P}(\Delta)}$. Therefore, suppose $p(\cdot)$ is the logical statement that defines the set $g(\Omega)$ for some $\Omega \subseteq \mathscr{P}(\Delta)$, we have

$$\Delta \setminus S \in \widehat{g(\Omega)} \iff S \in g(\Omega) \iff \Delta \setminus S \in g'\left(\widehat{\Omega}\right),$$

which concludes the first equality. The second equality follows directly since $g''(\cdot) = g(\cdot)$ by the definition of dual pair. □

Theorem 1. *Given any $\Omega \subseteq \mathscr{P}(\Delta)$, we have the following properties,*

$\mathscr{A}$. For the complement operator $\overline{(\cdot)}$,

- *$\mathscr{A}$1. $\overline{\varnothing} = \mathscr{P}(\Delta)$ and $\overline{\mathscr{P}(\Delta)} = \varnothing$*
- *$\mathscr{A}$2. Ω and $\overline{\Omega}$ form a partition of $\mathscr{P}(\Delta)$*
- *$\mathscr{A}$3. $\Omega \subseteq \Omega'$ if and only if $\overline{\Omega} \supseteq \overline{\Omega'}$ (order-reversing)*
- *$\mathscr{A}$4. Ω is an upper (lower) system implies $\overline{\Omega}$ is a lower (upper) system*
- *$\mathscr{A}$5. $\overline{\overline{\Omega}} = \Omega$ (self-inverse)*

$\mathscr{B}$. For the element-complement operator $\widehat{(\cdot)}$,

- *$\mathscr{B}$1. $\widehat{\varnothing} = \varnothing$ and $\widehat{\mathscr{P}(\Delta)} = \mathscr{P}(\Delta)$*
- *$\mathscr{B}$2. $\Omega \subseteq \Omega'$ if and only if $\widehat{\Omega} \subseteq \widehat{\Omega'}$ (order-preserving)*
- *$\mathscr{B}$3. $\widehat{(\cdot)}$ acts as a contravariant functor on Ω, i.e., for every $T, T' \in \Omega$ such that $T \subseteq T'$, we have $\Delta \setminus T, \Delta \setminus T' \in \widehat{\Omega}$ with inclusion direction reversed*

$\mathscr{B}4$. *Ω is an upper (lower) system implies $\widehat{\Omega}$ is a lower (upper) system*

$\mathscr{B}5$. $\widehat{\widehat{\Omega}} = \Omega$ *(self-inverse)*

$\mathscr{C}$. *For the cut operator $\mathcal{C}(\cdot)$,*

$\mathscr{C}1$. $\mathcal{C}(\varnothing) = \mathscr{P}(\Delta)$ *and* $\mathcal{C}(\mathscr{P}(\Delta)) = \varnothing$

$\mathscr{C}2$. $\mathcal{C}(\Omega) = \varnothing$ *if and only if* $\varnothing \in \Omega$

$\mathscr{C}3$. $\mathcal{C}(\Omega)$ *is an upper system*

$\mathscr{C}4$. $\mathcal{C}(\Omega) = \mathcal{C}(m(\Omega))$

$\mathscr{C}5$. $\Omega \subseteq \Omega'$ *implies* $\mathcal{C}(\Omega) \supseteq \mathcal{C}(\Omega')$ *(order-reversing)*

$\mathscr{C}6$. $\mathcal{C}(\mathcal{C}(\Omega)) = \uparrow \Omega$ *(upper envelope)*

$\mathscr{D}$. *For the cocut operator $\mathcal{G}(\cdot)$,*

$\mathscr{D}1$. $\mathcal{G}(\varnothing) = \mathscr{P}(\Delta)$ *and* $\mathcal{G}(\mathscr{P}(\Delta)) = \varnothing$

$\mathscr{D}2$. $\mathcal{G}(\Omega) = \varnothing$ *if and only if* $\Delta \in \Omega$

$\mathscr{D}3$. $\mathcal{G}(\Omega)$ *is a lower system*

$\mathscr{D}4$. $\mathcal{G}(\Omega) = \mathcal{G}(M(\Omega))$

$\mathscr{D}5$. $\Omega \subseteq \Omega'$ *implies* $\mathcal{G}(\Omega) \supseteq \mathcal{G}(\Omega')$ *(order-reversing)*

$\mathscr{D}6$. $\mathcal{G}(\mathcal{G}(\Omega)) = \downarrow \Omega$ *(lower envelope)*

$\mathscr{E}$. *For interactions between operators,*

$\mathscr{E}1$. $\overline{\widehat{\Omega}} = \widehat{\overline{\Omega}}$ *(commutative)*

$\mathscr{E}2$. $\uparrow \widehat{\Omega} = \widehat{\downarrow \Omega}$ *and* $\downarrow \widehat{\Omega} = \widehat{\uparrow \Omega}$ *(anticommutative)*

$\mathscr{E}3$. $\widehat{M(\Omega)} = m\left(\widehat{\Omega}\right)$ *and* $\widehat{m(\Omega)} = M\left(\widehat{\Omega}\right)$ *(anticommutative)*

$\mathscr{E}4$. $\mathcal{C}(\widehat{\Omega}) = \widehat{\mathcal{G}(\Omega)}$ *and* $\mathcal{G}(\widehat{\Omega}) = \widehat{\mathcal{C}(\Omega)}$ *(anticommutative)*

$\mathscr{E}5$. $\overline{\uparrow \Omega} \subseteq \downarrow \overline{\Omega}$ *and* $\overline{\downarrow \Omega} \subseteq \uparrow \overline{\Omega}$ *(partially anticommutative)*

$\mathscr{E}6$. $m\left(\overline{\Omega}\right) \subseteq \overline{M(\Omega)}$ *and* $M\left(\overline{\Omega}\right) \subseteq \overline{m(\Omega)}$ *(partially anticommutative)*

$\mathscr{E}7$. $\mathcal{G}\left(\overline{\Omega}\right) \subseteq \overline{\mathcal{C}(\Omega)}$ *and* $\mathcal{C}\left(\overline{\Omega}\right) \subseteq \overline{\mathcal{G}(\Omega)}$ *with the equalities hold if and only if Ω is an upper system and a lower system, respectively. (partially anticommutative)*

Proof. For the complement operator, we only prove ($\mathscr{A}4$) as the rest are directly from the definition. Suppose Ω is an upper system and take any $T \in \overline{\Omega}$. For every $T' \subsetneq T$, suppose $T' \notin \overline{\Omega}$ implies $T' \in \Omega$ since Ω and $\overline{\Omega}$ form a partition. But Ω is an upper system, which means $T \in \Omega$, a contradiction. The lower system case of Ω can be proved in the same fashion.

All statements regarding the element-complement operator are trivial, so we omit their proofs.

For the cut operator, we only prove ($\mathscr{C}6$). The rest are direct consequences of the definition. If $\varnothing \in \Omega$, then by the first two properties, $\mathcal{C}(\mathcal{C}(\Omega)) = \mathscr{P}(\Delta)$, which is equal to $\uparrow \Omega$ since $\varnothing \in \Omega$. Suppose $\varnothing \notin \Omega$, we have the following derivation,

$$U \in \mathcal{C}(\mathcal{C}(\Omega)) \Longleftrightarrow \forall S \in \mathcal{C}(\Omega),\ |U \cap S| \geq 1 \Longleftrightarrow U \in \uparrow \Omega,$$

where the first equivalence is the definition of the operator $\mathcal{C}$. For the "$\Leftarrow$" direction of the second equivalence, $U \in \uparrow \Omega$ means $U \supseteq U'$ for some $U' \in \Omega$. By definition, for all $S \in \mathcal{C}(\Omega)$, $|S \cap U'| \geq 1$, then this is also true for the superset U. For "$\Rightarrow$", we prove the contrapositive. Given $U \notin \uparrow \Omega$ and $\varnothing \notin \Omega$ by assumption, then for every $U' \in \Omega$, $U' \setminus U \neq \varnothing$. Then, $S = \bigcup_{U' \in \Omega} U' \setminus U$ intersects all elements in Ω, but $S \cap U = \varnothing$ by construction, contradicting the assumption that $|S \cap U| \geq 1$ for all $S \in \mathcal{C}(\Omega)$.

We prove the interactions before analyzing the cocut operation. For ($\mathscr{E}1$), we have

$$T \in \widehat{\overline{\Omega}} \Longleftrightarrow \Delta \setminus T \in \overline{\Omega} \Longleftrightarrow \Delta \setminus T \notin \Omega \Longleftrightarrow T \notin \widehat{\Omega} \Longleftrightarrow T \in \overline{\widehat{\Omega}},$$

where each equivalence is from the definitions. Statements ($\mathscr{E}2$)–($\mathscr{E}4$) are true due to Lemma 1. For ($\mathscr{E}5$), we have

$$T \in \overline{\uparrow \Omega} \Longleftrightarrow T \notin \uparrow \Omega \Longrightarrow T \notin \Omega \Longleftrightarrow T \in \overline{\Omega} \Longrightarrow T \in \downarrow \overline{\Omega}.$$

The second part can be proved by applying the complement operator on both sides. For ($\mathscr{E}6$), we clearly have the following

$$T \in m(\overline{\Omega}) \Longrightarrow T \notin \Omega \Longrightarrow T \notin M(\Omega) \Longrightarrow T \in \overline{M(\Omega)}.$$

The other half can be shown similarly. For ($\mathscr{E}7$), take any $T \in \mathcal{G}(\overline{\Omega})$, then for every $T' \notin \Omega$, we have $T \cup T' \neq \Delta$ by definition, which further implies that $\Delta \setminus T \in \Omega$. This shows that $T \notin \mathcal{C}(\Omega)$ since $T \cap (\Delta \setminus T) = \varnothing$. Thus, we have $\mathcal{G}(\overline{\Omega}) \subseteq \overline{\mathcal{C}(\Omega)}$. For the equality part, we first show the sufficiency. Take any $T \notin \mathcal{C}(\Omega)$, there exists some $T' \in \Omega$ such that $T' \cap T = \varnothing$. Suppose $T \notin \mathcal{G}(\overline{\Omega})$, then there exists some $T'' \notin \Omega$ such that $T \cup T'' = \Delta$. In particular, we have $T' \subseteq T''$. But, $T' \in \Omega$ and $T'' \notin \Omega$ contradicts that Ω is an upper system. For the necessity, we prove the contrapositive: suppose Ω is not an upper system, then $\overline{\mathcal{C}(\Omega)} \cap \overline{\mathcal{G}(\overline{\Omega})} \neq \varnothing$. Note that Ω is not an upper system implies that there exist $T \in \Omega$ and $T' \notin \Omega$ such that $T \subseteq T'$. We show that every T'' sandwiched between T and T' ensures that its complement $\Delta \setminus T''$ belongs to the intersection $\overline{\mathcal{C}(\Omega)} \cap \overline{\mathcal{G}(\overline{\Omega})}$ by the following

$$T \subseteq T'' \Longleftrightarrow T \subseteq \Delta \setminus (\Delta \setminus T'') \Longrightarrow T \cap (\Delta \setminus T'') = \varnothing \Longrightarrow \Delta \setminus T'' \notin \mathcal{C}(\Omega),$$

$$T' \supseteq T'' \Longleftrightarrow T' \supseteq \Delta \setminus (\Delta \setminus T'') \Longrightarrow T' \cup (\Delta \setminus T'') = \Delta \Longrightarrow \Delta \setminus T'' \notin \mathcal{G}(\overline{\Omega}),$$

which completes the proof of necessity. The other half of the statement can be proved by applying the complement operator and replacing Ω with $\overline{\Omega}$ on both sides.

Finally, for the cocut operator $\mathcal{G}(\cdot)$, we prove them by the previously established rules. For ($\mathscr{D}$1), we have

$$\mathcal{G}(\varnothing) = \widehat{\mathcal{C}(\widehat{\varnothing})} = \widehat{\mathcal{C}(\varnothing)} = \widehat{\mathscr{P}(\Delta)} = \mathscr{P}(\Delta),$$

where the other half can be proved similarly. For ($\mathscr{D}$2), suppose $\Delta \in \Omega$ then $\varnothing \in \widehat{\Omega}$, then $\mathcal{C}(\widehat{\Omega}) = \varnothing$ and $\mathcal{G}(\Omega) = \widehat{\mathcal{C}(\widehat{\Omega})} = \widehat{\varnothing} = \varnothing$. Property ($\mathscr{D}$3) is obvious as $\mathcal{G}(\Omega) = \widehat{\mathcal{C}(\widehat{\Omega})}$ and $\mathcal{C}(\cdot)$ returns an upper system. For ($\mathscr{D}$4), we have

$$\begin{aligned}
&\mathcal{C}(\widehat{\Omega}) = \mathcal{C}(m(\widehat{\Omega})) \Longrightarrow \\
&\mathcal{G}(\Omega) = \widehat{\mathcal{C}(\widehat{\Omega})} = \widehat{\mathcal{C}(m(\widehat{\Omega}))} = \mathcal{G}\left(\widehat{m(\widehat{\Omega})}\right) = \mathcal{G}\left(M\left(\widehat{\widehat{\Omega}}\right)\right) = \mathcal{G}(M(\Omega)),
\end{aligned}$$

where the last equality is due to ($\mathscr{B}$5).

For ($\mathscr{D}$5), $\Omega \subseteq \Omega'$ implies $\widehat{\Omega} \subseteq \widehat{\Omega}'$. Thus, $\mathcal{C}(\widehat{\Omega}) \supseteq \mathcal{C}(\widehat{\Omega}')$, which further implies $\widehat{\mathcal{C}(\widehat{\Omega})} \supseteq \widehat{\mathcal{C}(\widehat{\Omega}')}$ since the element-complement operator is order-preserving. This concludes the proof. Finally, for ($\mathscr{D}$6), we have

$$\mathcal{G}(\mathcal{G}(\Omega)) = \mathcal{G}\left(\widehat{\mathcal{C}(\widehat{\Omega})}\right) = \mathcal{C}\left(\widehat{\widehat{\widehat{\mathcal{C}(\widehat{\Omega})}}}\right) = \widehat{\mathcal{C}(\mathcal{C}(\widehat{\Omega}))} = \widehat{\uparrow \widehat{\Omega}} = \downarrow \Omega,$$

where the last equality is from ($\mathscr{E}$2) and ($\mathscr{B}$5). ☐

Corollary 1. *We have the following identities:*

- $\mathcal{G}(\cdot) = \widehat{\mathcal{C}\left(\widehat{(\cdot)}\right)}$;
- $\mathcal{E}(\cdot) = \widehat{\mathcal{C}(\cdot)} = \mathcal{G}\left(\widehat{(\cdot)}\right)$;
- $\mathcal{E}^2(\cdot) = \mathcal{G} \circ \mathcal{C}(\cdot)$;
- $\mathcal{G} \circ \mathcal{C}(\Omega) = \mathcal{P}(\Delta)$ *whenever* $\varnothing \in \Omega$, *and equals* $\varnothing$ *otherwise;*
- $\mathcal{C} \circ \mathcal{G}(\Omega) = \mathcal{P}(\Delta)$ *whenever* $\Delta \in \Omega$, *and equals* $\varnothing$ *otherwise.*

Proof. The first statement is due to ($\mathscr{E}$4) and ($\mathscr{B}$5). The second statement is true by the definition $\mathcal{E}$ and ($\mathscr{E}$4). The third statement can be proved by composing the two definitions of $\mathcal{E}$ from the second statement along with ($\mathscr{B}$5). For the fourth statement, note that $\mathcal{C}(\cdot)$ is always an upper system by ($\mathscr{C}$3), thus it either contains Δ or equals $\varnothing$. The former returns $\varnothing$ by ($\mathscr{D}$2) and the latter returns $\mathscr{P}(\Delta)$ by ($\mathscr{D}$1). Moreover, by ($\mathscr{C}$2), $\mathcal{C}(\Omega) = \varnothing$ if and only if $\varnothing \in \Omega$. The last statement can be proved similarly. ☐

Lemma 2 (Upper Embedding)**.** *For any set system* $\Omega \subseteq \mathscr{P}(\Delta)$*, we have*

$$\mathcal{C}\left(\widehat{\widehat{\Omega}}\right) \subseteq \Omega.$$

Moreover, this equality holds if and only if Ω is an upper system.

Proof. For the first statement, take any $T \in \mathcal{C}\left(\widehat{\overline{\Omega}}\right)$ and suppose $T \notin \Omega$. By definition, we have $T \in \overline{\Omega}$ and $\Delta \setminus T \in \widehat{\overline{\Omega}}$. On the other hand, $T \in \mathcal{C}\left(\widehat{\overline{\Omega}}\right)$ means $T \cap S \neq \varnothing$ for all $S \in \widehat{\overline{\Omega}}$. In particular, $T \cap (\Delta \setminus T) \neq \varnothing$, a contradiction. For the second statement, suppose Ω is an upper system, we take any $T \in \Omega$ such that $T \notin \mathcal{C}\left(\widehat{\overline{\Omega}}\right)$. Then, there exists some $S \in \widehat{\overline{\Omega}}$ such that $T \cap S = \varnothing$, which is equivalent to $T \subseteq \Delta \setminus S$. Note that $S \in \widehat{\overline{\Omega}}$ also means $\Delta \setminus S \in \overline{\Omega}$ by definition. Because Ω is an upper system, we know $\overline{\Omega}$ is a lower system, which implies $T \in \overline{\Omega}$, a contradiction. For the other direction, suppose $\Omega \subseteq \mathcal{C}\left(\widehat{\overline{\Omega}}\right)$, we show that Ω must be an upper system. Otherwise, there exists some $T \in \Omega$ and $T' \supseteq T$ such that $T' \notin \Omega$. Then, $T' \in \overline{\Omega}$ and $\Delta \setminus T' \in \widehat{\overline{\Omega}}$. Since $T \in \mathcal{C}\left(\widehat{\overline{\Omega}}\right)$ by assumption, we have $T \cap (\Delta \setminus T') \neq \varnothing$, which contradicts the choice of T'. This completes the proof. □

Theorem 2 (Upper Approximation)**.** *In the same problem setting, we have*

$$\mathcal{C}\left(\widehat{\overline{\Omega}}\right) = \mathcal{C}\left(\widehat{M(\overline{\Omega})}\right) = \mathcal{C}\left(m\left(\widehat{\overline{\Omega}}\right)\right) \subseteq \Omega \subseteq \mathcal{C}^2(\Omega) = \uparrow \Omega = \mathcal{C}\left(\widehat{\overline{\uparrow \Omega}}\right),$$

where $\mathcal{C}\left(\widehat{\overline{\Omega}}\right)$ and $\mathcal{C}\left(\widehat{\overline{\uparrow \Omega}}\right)$ are the tightest upper embedding and enclosing of Ω, respectively. Moreover, the equality holds for all if and only if Ω is an upper set.

Proof. The first two equalities are obvious since we have

$$\widehat{M(\overline{\Omega})} = m\left(\widehat{\overline{\Omega}}\right) \subseteq \widehat{\overline{\Omega}}$$

by ($\mathscr{E}$3), and the equality follows ($\mathscr{C}$4) and ($\mathscr{C}$5). The relation $\Omega \subseteq \mathcal{C}^2(\Omega) = \uparrow \Omega$ is a direct consequence of ($\mathscr{C}$6). The other inclusion is implied by Lemma 2 and ($\mathscr{C}$4). To prove $\mathcal{C}\left(m\left(\widehat{\overline{\Omega}}\right)\right)$ is the tightest embedding, consider any upper set $\Omega' \subseteq \Omega$, we show that

$$\Omega' \subseteq \mathcal{C}\left(\widehat{M(\overline{\Omega})}\right).$$

Suppose otherwise, take $T \in \Omega' \subseteq \Omega$ but $T \notin \mathcal{C}\left(\widehat{M(\overline{\Omega})}\right)$. The latter implies that for some $S \in \widehat{M(\overline{\Omega})}$, we have $T \subseteq \Delta \setminus S \in M(\overline{\Omega})$, where the membership also implies $\Delta \setminus S \notin \Omega$. However, since Ω' is an upper set, we have $\Delta \setminus S \in \Omega' \subseteq \Omega$, leading to the desired contradiction. For $\uparrow \Omega$, it is obvious from the definition that it is the tightest upper enclosing of Ω. Furthermore, the two inclusions become equalities when Ω is an upper system by Lemma 2 and ($\mathscr{C}$6). Finally, for the last equality, we use the relation

$$\mathcal{C}\left(\widehat{\overline{\Omega}}\right) \subseteq \Omega$$

and replace Ω with the upper set $\uparrow \Omega$, in which case the equality holds. □

Corollary 3. *Given a set system Ω, the tightest upper embeddings with respect to $\overline{\Omega}$, $\widehat{\Omega}$, and $\widehat{\overline{\Omega}}$ are*

$$\mathcal{C}(\widehat{\Omega}) \subseteq \overline{\Omega},\ \mathcal{C}(\overline{\Omega}) \subseteq \widehat{\Omega},\ \mathcal{C}(\Omega) \subseteq \widehat{\overline{\Omega}},$$

and their equalities hold if and only if Ω is a lower system for the first two cases and is an upper set for the third case. Symmetrically, $\mathcal{G}(\widehat{\Omega})$, $\mathcal{G}(\overline{\Omega})$, and $\mathcal{G}(\Omega)$ are the respective tightest lower embeddings with equality conditions reversed.

Proof. Replacing Ω in the main result of Lemma 2 with each of the right-side sets, we can derive these claims using the commutativity rule ($\mathscr{E}1$) and the self-inverse rules ($\mathscr{A}5$) and ($\mathscr{B}5$). They are the tightest upper embeddings due to Theorem 2 □

Lemma 3. *Given any set system Ω, the vector representations of $\mathcal{C}(\Omega)$ and $\mathcal{E}(\Omega) = \mathcal{G}(\widehat{\Omega})$ are equivalent to the following* covering *and* elimination *inequalities, respectively.*

$$\mathcal{X}_{\mathcal{C}(\Omega)} = \left\{ x \in \{0,1\}^n \;\middle|\; \sum_{i \in T} x_i \geq 1, \forall T \in \Omega \right\},$$

$$\mathcal{X}_{\mathcal{E}(\Omega)} = \left\{ x \in \{0,1\}^n \;\middle|\; \sum_{i \in T} x_i \leq |T| - 1, \forall T \in \Omega \right\}.$$

Moreover, these two types of inequalities can be equivalently converted to each other by the substitution $y := 1 - x$.

Proof. By definition, $S \in \mathcal{C}(\Omega)$ if and only if $S \cap T \neq \varnothing$ for all $T \in \Omega$. Thus, the vector representation x_S is feasible if and only if it satisfies all the covering inequalities. On the other hand, we have $S \in \mathcal{E}(\Omega) = \widehat{\mathcal{C}(\Omega)}$ if and only if $\Delta \setminus S \in \mathcal{C}(\Omega)$. Hence, vector $y_S := 1 - x_S$ indicates the complement structure $\Delta \setminus S$ and intersects every $T \in \Omega$. Substituting $x = 1 - y$ in the covering inequalities produces the elimination constraints. In particular, this also proves the last claim. □

Theorem 3. *Any general BIP (2) can be reformulated into the following forms for inner/outer approximations,*

Upper Embedding Π_{ui}:

$$\min_{x \in \{0,1\}^n} \ f(x) \tag{3a}$$

$$s.t. \ \sum_{i \in T} x_i \geq 1,\ \forall T \in m\left(\widehat{\overline{\Omega}}\right) \tag{3b}$$

Upper Enclosing Π_{uo}:

$$\min_{x \in \{0,1\}^n} \ f(x) \tag{4a}$$

$$s.t. \ \sum_{i \in T} x_i \geq 1,\ \forall T \in m\left(\widehat{\overline{\uparrow \Omega}}\right) \tag{4b}$$

Lower Embedding Π_{li}:

$$\min_{x \in \{0,1\}^n} \ f(x) \tag{5a}$$

$$s.t. \ \sum_{i \in T} x_i \leq |T| - 1,\ \forall T \in m\left(\overline{\Omega}\right) \tag{5b}$$

Lower Enclosing Π_{lo}:

$$\min_{x \in \{0,1\}^n} \ f(x) \tag{6a}$$

$$s.t. \ \sum_{i \in T} x_i \leq |T| - 1,\ \forall T \in m\left(\overline{\downarrow \Omega}\right) \tag{6b}$$

with objective values satisfying

$$z(\Pi_{uo}) \leq z(\Pi) \leq z(\Pi_{ui}), \quad z(\Pi_{lo}) \leq z(\Pi) \leq z(\Pi_{li}).$$

Moreover, (3) *and* (4) *are both equivalent to* (2) *if and only if* Ω *is an upper system;* (5) *and* (6) *are both equivalent to* (2) *if and only if* Ω *is a lower system.*

Proof. Theorem 2 shows that $\tilde{\mathcal{C}}(\Omega) \subseteq \Omega \subseteq \tilde{\mathcal{C}}(\uparrow \Omega)$ are the tightest inner and outer approximations of Ω. Then, the reformulations Π_{ui} and Π_{uo} follow directly from Lemma 3. Similarly, we have $\tilde{\mathcal{G}}(\Omega) = \mathcal{E}(\overline{\Omega})$, which leads to the reformulations Π_{li} and Π_{lo}. The rest of the claim follows directly by Theorem 2 and Corollary 2. ☐

Corollary 4. *The feasibility and redundancy conditions of the approximations are the following*

- Π_{ui} *is infeasible iff* $\Delta \notin \Omega$*, and is redundant iff* $\Omega = \mathscr{P}(\Delta)$*;*
- Π_{uo} *is infeasible iff* $\Omega = \varnothing$*, and is redundant iff* $\varnothing \in \Omega$*;*
- Π_{li} *is infeasible iff* $\varnothing \notin \Omega$*, and is redundant iff* $\Omega = \mathscr{P}(\Delta)$*;*
- Π_{lo} *is infeasible if and only if* $\Omega = \varnothing$*, and is redundant if and only if* $\Delta \in \Omega$*.*

Proof. We only prove the first statement since the rest can be shown similarly. Π_{ui} is infeasible and redundant whenever its solution space's set system representation $\mathcal{C}\left(\widehat{\overline{\Omega}}\right)$ equals $\varnothing$ and $\mathscr{P}(\Delta)$, respectively. The former occurs exactly when $\varnothing \in \widehat{\overline{\Omega}}$ by ($\mathscr{C}$2), which leads to the characterization $\Delta \notin \Omega$. The latter happens if and only if $\widehat{\overline{\Omega}} = \varnothing$ according to ($\mathscr{C}$1) and ($\mathscr{C}$6). This leads to the condition $\Omega = \mathscr{P}(\Delta)$. ☐

Theorem 4. *Given an upper system* $\Omega \neq \varnothing$ *as the solution space of* (3)*, a structure* $T_x \subseteq \Delta$*, and a chain* C *from* T_x *to* Δ*. Algorithm 1 returns* $\varnothing$ *whenever* T_x *is feasible, and returns a structure in* $\widehat{\overline{\Omega}}$ *associated with the maximum element in* $C \cap \overline{\Omega}$ *if otherwise. Let* $O(\tau(\Omega))$ *be the complexity of the membership oracle* $\mathbb{I}_\Omega$*, the complexity for separating such an inequality in* (3b) *is* $O(\log(|\Delta| - |T_x| + 1) \cdot \tau(\Omega))$*.*

Proof. Constraint (3b) is equivalent to

$$\sum_{i \in \Delta \setminus S} x_i \geq 1, \ \forall S \in \overline{\Omega}.$$

Given the current solution T_x, we need to separate some element in $\widehat{\overline{\Omega}}$ that are not intersected by T_x, i.e., some $S \in \overline{\Omega}$ such that

$$T_x \cap (\Delta \setminus S) = \varnothing \Longleftrightarrow S \supseteq T_x.$$

Thus, we need to separate S from some chain C from T_x to Δ. By design, Algorithm 1 returns $\varnothing$ if and only if $\mathbb{I}_\Omega(T_x) = 1$, i.e., when $T_x \in \Omega$. Since Ω is an upper system, there is no $S \supseteq T_x$ that belongs to $\overline{\Omega}$. Equivalently speaking, every structure in $\widehat{\overline{\Omega}}$ has been intersected by T_x, i.e., T_x is a feasible solution. Now, suppose $\mathbb{I}_\Omega(T_x) = 0$, every inclusion chain $C = (T_x, T_1, \ldots, \Delta)$ can be split at some T_j such that every set in $(T_x, T_1, \ldots, T_j)$ belongs to $\overline{\Omega}$ and every set in $(T_{j+1}, \ldots, \Delta)$ belongs to Ω since Ω is upward-closed. Thus, the proposed binary search will return the maximum element in $C \cap \overline{\Omega}$. Finally, the main time complexity of Algorithm (1) resides in the binary search subroutine, which requires at most $\log(|\Delta| - |T_x| + 1)$ inquiries. Moreover, each inquiry is dominated by the verification complexity $O(\tau(\Omega))$, thus gives the claimed runtime complexity. ☐

Lemma 4. *Given any binary solution space $\mathcal{X} := \{x \in \{0,1\}^n \mid g(x) \le 0\}$, define*

$$\widehat{\mathcal{X}} := \{x \in \{0,1\}^n \mid g(1-x) \le 0\}.$$

Then, the function $\theta(x) = 1 - x$ establishes a bijective affine transformation from $\operatorname{conv}(\mathcal{X})$ to $\operatorname{conv}(\widehat{\mathcal{X}})$. In particular, an inequality $\langle a, x \rangle \ge b$ is facet-defining for $\operatorname{conv}(\mathcal{X})$ if and only if $\langle a, \theta(x) \rangle \ge b$ is facet-defining for $\operatorname{conv}(\widehat{\mathcal{X}})$.

Proof. The transformation $\theta(\cdot)$ is clearly affine and injective by definition. Moreover, $x \in \operatorname{conv}(\mathcal{X})$ if and only if $x = \sum_{j \in [m]} \lambda_j x_j$ for some convex combination coefficients λ_j and $x_j \in \mathcal{X}$. Equivalently, we have

$$\theta(x) = 1 - x = \sum_{j \in [m]} \lambda_j (1 - x_j) = \sum_{j \in [m]} \lambda_j \theta(x_j) \Longleftrightarrow \theta(x) \in \operatorname{conv}(\widehat{\mathcal{X}}).$$

Thus, θ is an affine bijection. In particular, it preserves all the extreme points as well as affine independence. Suppose $\langle a, x \rangle \ge b$ is a facet of $\operatorname{conv}(\mathcal{X})$, there are n affinely independent extreme points $\{x_j\}_{j \in [n]} \subseteq \operatorname{conv}(\mathcal{X})$ that are tight on the inequality. Then, $\theta(x_j)$'s form n affinely independent extreme points of $\widehat{\mathcal{X}}$ that are tight on the inequality $\langle a, \theta(x) \rangle \ge b$, which concludes the proof. ☐

Theorem 6. *Suppose Ω is an upper set and $|T| \ge 2$ for every $T \in m(\widehat{\overline{\Omega}})$, the covering inequality (3b) is facet-defining if and only if the associated maximal non-solution $\Delta \setminus T \in M(\overline{\Omega})$ is quasi-feasible. Symmetrically, suppose Ω is a lower set and $|T| \ge 2$ for every $T \in m(\overline{\Omega})$, the elimination inequality (5b) is facet-defining if and only if the associated minimal non-solution $T \in m(\overline{\Omega})$ is co-quasi-feasible.*

Proof. According to Proposition 3.4 in [36], given Ω is an upper set and $|T| \ge 2$ for every $T \in m(\widehat{\overline{\Omega}})$, (3b) is facet-defining if and only if for every $a \in \Delta \setminus T$ there exists some $a' \in T$ such that $T \setminus \{a'\} \cup \{a\} \notin \widehat{\overline{\Omega}}$, that is, $\Delta \setminus (T \setminus \{a'\} \cup \{a\}) = (\Delta \setminus T) \setminus \{a\} \cup \{a'\} \in \Omega$. By definition, this means $\Delta \setminus T$ is quasi-feasible. For the second statement, we have

$$\sum_{i \in T} x_i \le |T| - 1 \Longleftrightarrow \sum_{i \in T} (1 - x_i) \ge 1.$$

By Lemma 4, we only need to investigate the facet-defining conditions for the inequality

$$\sum_{i \in T} x_i \geq 1$$

in the isomorphic polyhedron $\widehat{\mathcal{X}}$. Then, by the same proposition in [36], given $|T| \geq 2$ for all $T \in m(\overline{\Omega})$, this inequality is facet-defining if and only if for every $a' \in \Delta \setminus T$ there exists some $a \in T$ such that $T \setminus \{a\} \cup \{a'\} \in \Omega$, i.e., T is co-quasi-feasible. □

Proposition 2. *Given $G = (V, E)$ with $|V| \geq 3$, $T \in \overline{\uparrow \Omega}$ if T satisfies any of the following conditions,*

- *$G(T)$ is disconnected;*
- *for some $v \in V(G(T))$, $deg(v) = 1$;*
- *there exists a vertex bipartition such that a non-singleton part has at most one vertex connected to the other part;*
- *$G(T) = cl(G(T'))$ for any above T'.*

Proof. The first three cases are trivial to verify, and the last one is by the Bondy-Chvátal Theorem. □

Proposition 3. *Given $G = (V, E)$, $T \in \overline{\downarrow \Omega}$ if and only if T satisfies any of the following conditions,*

- *T contains a none-Hamiltonian cycle (NHC);*
- *T contains a claw (i.e., a star with three edges);*
- *T is a set of mutually disjoint paths that cannot be extended to any HC.*

Moreover, when G is a complete graph, the first two types constitute the entire $\overline{\downarrow \Omega}$.

Proof. Clearly, no NHC nor claw can be contained by any HC. Any other type of subgraph must consist of mutually disjoint paths, which can be trivially extended to an HC in a complete graph. □

Proposition 4. *The following functions are entry-wise monotone:*

- *Monotone functions.*
- *Linear (modular) functions.*
- *For every $i \in \Delta$, $\partial_i f([0,1]^n)$ is entirely contained in $\mathbb{R}_+$ or $\mathbb{R}_-$.*
- *Submodular functions f where for every $i \in \Delta$, either*

$$f(\{i\}) - f(\varnothing) \leq 0 \text{ or } f(\Delta) - f(\Delta \setminus \{i\}) \geq 0.$$

- *Subadditive functions f where $f(e_i) \le 0$ for all $i \in \Delta$.*
- *Supermodular and superadditive counterparts.*

Proof. The first case is trivial. Every linear function $\langle c, x\rangle$ satisfies $\Delta_{x_{-i}} = c_i$. Thus, the sign will not change by picking a different x_{-i}. The third conditions ensure that $\partial_i f(\cdot, x_{-i})$ is a monotone function on the interval $[0, 1]$ regardless of the value of x_{-i}. For submodular function, we have

$$f(\{i\}) - f(\varnothing) \ge f(T \cup \{i\}) - f(T) \ge f(\Delta) - f(\Delta \setminus \{i\})$$

for every $i \in \Delta$ and every $T \subseteq \Delta \setminus \{i\}$. Suppose the conditions in the claim are satisfied, the sign of $\Delta_{x_{-i}}$ will never change. For subadditive functions, we have

$$f(1, x_{-i}) = f((1, 0_{-i}) + (0, x_{-i})) \le f(1, 0_{-i}) + f(0, x_{-i}) \Longrightarrow \Delta_{x_{-i}} \le f(e_i) \le 0.$$

Thus, changing x_{-i} will not affect the sign. □

Proposition 5 (Optimality-Invariant Standardization II)**.** *For Problem (2) with an entry-wise monotone objective function, we can obtain an equivalent reformulation with an increasing (or decreasing) function by setting $x_i = 1 - x'_i$ for all $i \in I := \{i \in \Delta \mid \Delta_{x_{-i}} < 0\}$ (or $I := \{i \in \Delta \mid \Delta_{x_{-i}} > 0\}$).*

Proof. First, this substitution will not change the objective value, and the optimal solution can be recovered by $x_i = 1 - x'_i$. Thus, this reformulation is equivalent. To show the resulting objective function $\hat{f}$ is increasing, we have

$$\begin{aligned} &\forall i \in I,\ \hat{f}(1, x'_{-i}) - \hat{f}(0, x'_{-i}) = f(0, x_{-i}) - f(1, x_{-i}) = -\Delta_{x_{-i}} > 0,\ \forall x'_{-i} \in \{0,1\}^n \\ &\forall i \notin I,\ \hat{f}(1, x'_{-i}) - \hat{f}(0, x'_{-i}) = f(1, x_{-i}) - f(0, x_{-i}) = \Delta_{x_{-i}} \ge 0,\ \forall x'_{-i} \in \{0,1\}^n. \end{aligned}$$

Thus, adding element $i \in \Delta$ into a set $T \subseteq \Delta$ results in a nonnegative increase in the objective value. Then, for any $T \subseteq T'$, we can add elements $T' \setminus T$ into T one by one, each step of which adds a nonnegative value. The same argument is valid for the decreasing case. □

Lemma 5. *A set system Ω has the interval property if and only if it can be written as $\Omega = \Omega_1 \cap \Omega_2$ for some upper system Ω_1 and some lower system Ω_2.*

Proof. For necessity, we show that every interval set system Ω can be written as $\Omega = (\uparrow \Omega) \cap (\downarrow \Omega)$. Take any $T \in \Omega$, we have $T \in \uparrow \Omega$ as well as $T \in \downarrow \Omega$, which proves one inclusion. For the other direction, take any $T \in (\uparrow \Omega) \cap (\downarrow \Omega)$. By the definitions of upper and lower closures, there exist $T', T'' \in \Omega$ such that $T' \subseteq T \subseteq T''$. Then, the interval property of Ω ensures that $T \in \Omega$. For sufficiency, take any $T, T' \in \Omega$ such that $T \subseteq T'$. We have $T \in \Omega_1$ and $T' \in \Omega_2$. Take any T'' sandwiched between T and T', the upward closure of Ω_1 and the downward closure of Ω_2 enforce that $T'' \in \Omega_1 \cap \Omega_2 = \Omega$, which shows that Ω is an interval set system. □

Corollary 6. *Suppose the solution space Ω of Problem (2) is an interval set system, then the following is an exact reformulation of (2).*

$$\min_{x\in\{0,1\}^n} \ f(x) \tag{7a}$$

$$s.t. \ \sum_{i\in T} x_i \geq 1, \qquad \forall T \in m\left(\widehat{\uparrow \Omega}\right), \tag{7b}$$

$$\sum_{i\in T} x_i \leq |T| - 1, \ \forall T \in m\left(\overline{\downarrow \Omega}\right). \tag{7c}$$

Moreover, Algorithm 1 and 3 can be used to separate constraints in (7b) and (7c) with membership oracles for $\uparrow\Omega$ and $\downarrow\Omega$.

Proof. Lemma 3 implies (7b) and (7c) characterize the respective $\mathcal{C}\left(\widehat{\uparrow \Omega}\right)$ and $\mathcal{G}\left(\widehat{\downarrow \Omega}\right)$, which are equivalent to $\uparrow\Omega$ and $\downarrow\Omega$ by Theorem 2 and Corollary 2. Then, the claim is trivially true since the interval system $\Omega = (\uparrow\Omega) \cap (\downarrow\Omega)$ by Lemma 5. □

Theorem 8 (General Min-Cut Max-Flow Duality)**.** *For any cut pair Ω and $\widehat{\widehat{\Omega}}$ with a non-negative cost vector c assign to the ground set Δ, there exists a flow function f with c as the capacity vector such that*

$$\min_{x\in\mathcal{X}_{\widehat{\widehat{\Omega}}}} \langle c, x\rangle = \max_{x\in\mathcal{X}_{\Omega}} f(c,x), \tag{9}$$

and f is increasing on both c and x.

Proof. We have $\widehat{\widehat{\Omega}} = \mathcal{C}(\Omega) = \mathcal{C}(m(\Omega))$. Hence, the left side of (9) equals

$$\min_{S\in\mathcal{C}(m(\Omega))} \sum_{i\in S} c_i.$$

To finish the proof, we need to construct a max-flow problem using the structures in Ω and the capacity c such that the corresponding max-flow objective $f(c,x)$ equals the above min-cut value. We define this flow problem as follows (see Figure 2 for an illustration):

- Create two artificial nodes s and t and an artificial arc from t to s.
- Elements in every $T \in m(\Omega)$ is sorted by the order of elements in Δ.
- For each $T \in m(\Omega)$, we create a straight line l_T representing a flow pipe, then connect node s to the start point and link the endpoint to node t. Then, for each element $i \in T$ in order, we create a node on l_T from the start point to end, and consider them as some type of checkpoint on the pipe.
- For each $i \in \Delta$, if i labels multiple nodes in the above-constructed graph, we will merge all these nodes as one and leave the rest of the diagram unchanged. After this step, each $i \in \Delta$ should correspond to at most one node (checkpoint) in the graph.

- At each checkpoint i, the maximum combined flow capacity is assigned as c_i, and the capacities on all arcs are set to be $+\infty$.
- At each checkpoint i, the pipes passing i share the assigned capacity c_i but cannot cross the pipes, i.e., flows in different the pipes are separated.
- Finally, we define $f(c, x)$ as a function that returns the maximum combined flow in this max-flow problem where the checkpoint $i \in \Delta$ is open (for flow to pass) if and only if $x_i = 1$.

By this design, the function f is clearly increasing on c and x since a larger c provides more capacity on the checkpoints, and a larger x implies more checkpoints are open for flow to pass. Since Ω is an upper system by the definition of the cut pair, the right side of (9) equals $f(c, 1)$, i.e., the maximum flow in the constructed graph with capacities c and with all the checkpoints open. Moreover, this constructed graph ensures that every cut in $\mathcal{C}(\Omega)$ corresponds to a unique plan to disconnect all the pipes by blocking the corresponding checkpoints. Moreover, the maximum combined flow in the graph equals the bottleneck among all such cuts, which equals the value of min-cut. This concludes the proof. □