Set System Approximation for Binary Integer Programs: Reformulations and Applications

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

Abstract

Covering and elimination inequalities are central to combinatorial optimization, yet their role has largely been studied in problem-specific settings or via no-good cuts. This paper introduces a unified perspective that treats these inequalities as primitives for set system approximation in binary integer programs (BIPs). We show that arbitrary set systems admit tight inner and outer monotone approximations, exactly corresponding to covering and elimination inequalities. Building on this, we develop a toolkit that both recovers classical structural correspondences (e.g., paths vs. cuts, spanning trees vs. cycles) and extends polyhedral tools from set covering to general BIPs, including facet conditions and lifting methods. We also propose new reformulation techniques for nonlinear and latent monotone systems, such as auxiliary-variable-free bilinear linearization, bimonotone cuts, and interval decompositions. A case study on distributionally robust network site selection illustrates the framework's flexibility and computational benefits. Overall, this unified view clarifies inner/outer approximation criteria, extends classical polyhedral analysis, and provides broadly applicable reformulation strategies for nonlinear BIPs.

Keywords: monotone cuts, facet analysis, site selection

1 Introduction

Valid inequalities are central to combinatorial optimization, not only for strengthening formulations but also for revealing structural properties of feasible sets. Among them, two recurring families are the *covering* and *elimination* inequalities, defined for a structured subset T (e.g., paths, trees, cycles) as

$$\sum_{i\in T} x_i \geq 1 \quad \text{(Covering)}, \quad \sum_{i\in T} x_i \leq |T|-1 \quad \text{(Elimination)}.$$

These inequalities have been widely recognized, derived, and often strengthened into stronger forms across a broad range of combinatorial optimization problems, including transportation and routing problems [1, 9, 25, 32, 49], network design and survivability [27], facility location [14, 36], sequencing and scheduling models [16], and interdiction games [15, 31, 40, 53, 57].

Traditionally, such constraints are derived from problem-specific structures: for example, enforcing coverage over required sets leads to covering inequalities, while eliminating cycles in the traveling salesman problem (TSP) naturally yields elimination constraints. Subsequent frameworks such as Logic-Based Benders Decomposition (LBBD) [30] and Combinatorial Benders Decomposition (CBD) [17] have unified these ideas by interpreting such inequalities as *no-good cuts* derived from infeasible solutions under monotone objective functions. In these frameworks, covering and elimination inequalities are generated iteratively to refine the feasible region by excluding solutions encountered during the solution process.

Despite these advances, several important questions regarding the broader role of these inequalities remain open in the context of general binary solution spaces:

- When do covering and elimination inequalities yield valid outer or inner approximations of binary solution spaces, and under what conditions are these approximations exact?
- What general conditions ensure that such inequalities define facets of the convex hull of the feasible region?
- How to extend covering and elimination inequalities to fully characterize nonlinear feasible regions?

This paper addresses all three inquiries by introducing a new perspective that treats covering and elimination inequalities as primitive tools for *binary solution space approximation*, rather than as post hoc feasibility cuts. This shift enables a unified method that (i) recovers classical structural correspondences—such as s-t paths versus s-t cuts in flow problems [23], spanning trees versus edge cuts in min-cut problems [33], and edge cuts versus bipartite subgraphs in max-cut [7, 8]; (ii) extends classical tools from the set covering literature—including polyhedral analysis [4, 5, 51], lifting techniques [47, 56], and supervalid inequalities [57]—to general binary integer programs (BIPs); and (iii) introduces new reformulation tools by identifying latent *monotone systems*, i.e., solution spaces closed under subset or superset inclusion, thereby enabling more flexible use of these inequalities across diverse problem settings. In addition, we also introduce a compact set of algebraic properties of set system operators, extending the classical clutter–blocker relationship [18, 21] to a richer family of operators. This serves as a convenient toolkit for structural reasoning and simplifies several of our results.

1.1 Related Works

The covering and elimination inequalities are closely related to two prominent strands of research. The first concerns classical constraint families such as set covering inequalities [5] and subtour elimination constraints for TSP [9], which are widely used in mixed integer programming problems to strengthen formulations and improve computational efficiency. The second relates to no-good cuts and their generalizations within the frameworks of LBBD [30] and CBD [17], where valid inequalities are iteratively derived from infeasible solutions to refine the master problem. Both offer insights that motivate the development of our set system approximation framework.

Set covering inequalities are widely employed in different contexts, such as interdiction games [31, 40, 58], vehicle routing [49], network design [27], power grid optimization [59], and facilitate location problems [14], often enhancing the branch-and-cut implementation framework. Moreover, these covering inequalities have shown to possess strong facet properties in multiple problem settings [51, 52, 56]. Similarly, subtour elimination inequalities have become a critical reformulation component in transportation [1, 25, 32], production scheduling [16], location-routing problems [36], and have demonstrated significant impact in computational efficiency when properly strengthened and implemented [20, 54].

In many applications, covering and elimination inequalities are derived via Benders feasibility cuts [10], typically followed by strengthening steps to eliminate big-M constants. This led to the development of *combinatorial Benders decomposition* (CBD) [17], which generalizes Benders methods to mixed-integer problems with binary and continuous variables. A further generalization is offered by *logic-based Benders decomposition* (LBBD) [30], which accommodates nonlinear and combinatorial subproblems by deducing new constraints from infeasible subproblem outcomes via logic inference. Due to its generality, LBBD has found wide applications in areas such as transportation, production, supply chain management, and telecommunications [11, 12, 38, 46], with comprehensive discussions available in [29, 50]. A notable feature of both LBBD and CBD is the use of *no-good cuts*, which are typically defined for an encountered infeasible solution $x \in \{0,1\}^n$ with index set $I := \{i \in [n] \mid x_i = 1\}$ as

$$\sum_{i \in [n] \setminus I} x_i + \sum_{i \in I} (1 - x_i) \geq 1,$$

ensuring that at least one variable in x must flip to eliminate this infeasible solution from the feasible set. When the objective function exhibits monotonicity, stronger variants, termed *monotone cuts*, become valid. These take the form

$$\sum_{i \in [n] \setminus I} x_i \geq 1 \quad \text{or} \quad \sum_{i \in I} x_i \leq |I| - 1,$$

depending on whether the objective function is increasing or decreasing in a minimization setting.

While the constraints in our framework resemble monotone cuts in form, the underlying focus is quite different. LBBD, CBD, and no-good cuts mainly serve to dynamically generate valid inequalities from individual infeasible solutions to guide solver convergence. By contrast, our framework focuses on the structural approximation of the binary solution space itself. Rather than operating on encountered solution vectors, it analyzes combinatorial set systems deduced from the entire family of non-solutions. Depending on the intended approximation purposes—outer or inner—the resulting constraints need not be valid in the traditional logical sense; instead, they serve as structural primitives for representing or approximating the solution space via latent monotone subsystems, which may require additional reformulation or decomposition steps to uncover (e.g., bimonotone cuts or interval system decompositions in Section 4). Table 1 summarizes the key

Aspect	LBBD / CBD / No-Good Cuts	Proposed Framework
Goal	Refine the feasible set of an optimization model using valid cuts	Approximate and characterize feasible structures of a given binary space
Constraint Validity	Constraints must be logically valid for the original problem	Constraints may be invalid (especially for inner approximation) but serve structural representation
Practical Usage	Iteratively adds no-good cuts when infeasibility arises to guide convergence	Uncover latent monotone subsystems (e.g., bimonotone or interval systems) for approximation and reformulation
Methodology	Dynamically generate cuts from individual infeasible solutions encountered during solving	Statically analyze structures (e.g., set or graph families) deduced from the entire non-solution space

Table 1: Comparison between classical methods (LBBD, CBD, and no-good cuts) and the proposed set system approximation framework.

differences between the two approaches.

1.2 Contributions

Our contributions, organized around the three motivating inquiries in the introduction, are summarized below.

- **Set System Approximation.** We establish the tightest inner and outer approximations of arbitrary set systems by monotone systems (Theorem 1), and prove their exact correspondence with covering and elimination inequalities (Theorem 2). This resolves the *first inquiry* by showing precisely when such inequalities yield outer/inner approximations of binary solution spaces, and when these approximations become exact. In doing so, we also develop a compact cut–cocut algebra (Appendix A) that extends the classical clutter–blocker framework [18, 21] to a broader family of operators.

- **Unified Analytical Toolkit.** Building on this result, we develop a unified toolkit for analyzing binary integer programs (BIPs). First, we introduce a systematic method for identifying structural correspondences, recovering well-known structure pairs (e.g., cuts vs. paths, odd cycles vs. cuts, neighborhoods vs. dominating sets) directly from the framework (Section 3.1). Second, we extend classical polyhedral tools from set covering to general BIPs (Section 3.2), including a general facet-defining condition (Corollary 5). This addresses the *second inquiry*, demonstrating how existing combinatorial and polyhedral insights fit within the unified set-system perspective.

- **New Reformulations.** We introduce several reformulation methods for nonlinear and latent monotone systems that, to the best of our knowledge, are new. These include: (i) a linearization of bilinear terms $\langle x, Ry \rangle$ that avoids auxiliary variables when R is structured

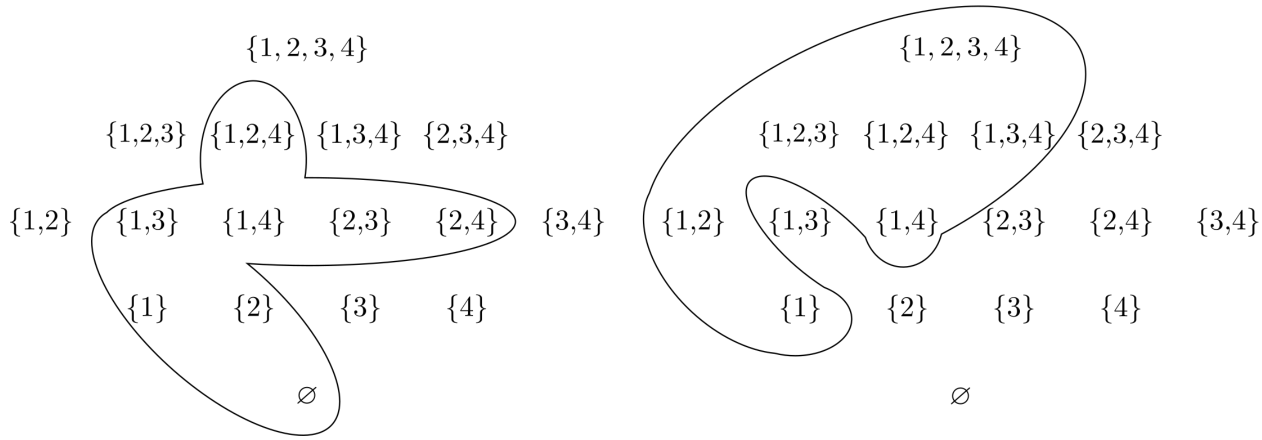

Figure 1: Two set systems, Ω_1 (left) and Ω_2 (right), each defined over the ground set $\Delta = \{1, 2, 3, 4\}$. Each curve encloses all subsets belonging to Ω_1 (left) or Ω_2 (right).

(Example 2), in contrast to the standard McCormick linearization [43]; (ii) bimonotone cuts, obtained by extending monotone cuts via the flipping map, which to our knowledge have not been exploited for exact reformulation, yielding tractable models such as the signed min-cut (Example 9); (iii) interval system decompositions, which provide reformulations for piecewise-affine objectives and disjunctive criteria (Examples 10–11). Collectively, these tools address the *third inquiry* and extend the scope of monotone reformulations beyond classical settings.

Throughout, we adopt the convention that if the feasible region is empty, the objective value is $+\infty$ for minimization and $-\infty$ for maximization. The remainder of the paper is organized as follows. Section 2 develops the main set system approximation results for binary solution spaces. Section 3 demonstrates how these results recover structural correspondences in classical combinatorial problems and extend polyhedral tools from the set covering literature to general BIPs. Section 4 introduces two complementary reformulation strategies based on latent monotone system identification. Section 5 presents a case study on network site selection under distributional uncertainty, highlighting the flexibility and efficiency of the proposed framework. Finally, Section 6 offers concluding remarks and a summary of our findings. Technical proofs are deferred to Appendix B to streamline exposition.

2 Set System Approximation

Every binary vector $x \in \{0, 1\}^n$ can be uniquely associated with a subset of the ground set $\Delta := [n]$, defined by $T_x := \{i \in \Delta \mid x_i = 1\}$. Conversely, for a subset $T \subseteq \Delta$, let x_T denote the corresponding binary vector. Under this correspondence, any binary solution space $\mathcal{X} \subseteq \{0, 1\}^n$ can be equivalently represented as a *set system* $\Omega_{\mathcal{X}} := \{T \subseteq \Delta \mid x_T \in \mathcal{X}\}$, which is a subset of the power set denoted as $\mathscr{P}(\Delta)$. Figure 1 illustrates two set systems with $n = 4$.

This section develops the tightest inner and outer approximations of a given set system Ω using simpler structures, termed the *monotone systems* defined as follows.

Definition 1 (Monotone Systems)**.** A set system $\Omega \subseteq \mathscr{P}(\Delta)$ is *upper-closed* if $T \in \Omega$ and $T' \supseteq T$ imply $T' \in \Omega$; it is *lower-closed* if $T \in \Omega$ and $T' \subseteq T$ imply $T' \in \Omega$. Any upper- or lower-system is called a *monotone system*.

It is well-known that any union or intersection of upper-systems (or lower-systems) are still upper-closed (or lower-closed). To support the main approximation results, we introduce a set of basic operators that transform one set system into another.

Definition 2. For a set system Ω, we define eight operators organized into two groups.

- Closure & Extremal Operators $\uparrow(\cdot), \downarrow(\cdot), m(\cdot), M(\cdot)$:
  - $\uparrow \Omega := \{T \mid T \supseteq T' \text{ for some } T' \in \Omega\}$ (*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*).
- Complement & Cut Operators $\overline{(\cdot)}, \widehat{(\cdot)}, \mathcal{C}(\cdot), \mathcal{G}(\cdot)$:
  - $\overline{\Omega} := \mathscr{P}(\Delta) \setminus \Omega$ (*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*),

The up- and down-closure operators return the smallest (with respect to inclusion) upper- or lower-systems containing Ω, respectively. Conversely, the minimal and maximal operators remove redundant elements by retaining only the minimal or maximal sets in a monotone system. The complement operator returns all subsets of Δ not in Ω (i.e., structures associated with all non-solutions), while the element-complement operator takes the complement of each $T \in \Omega$ within Δ (e.g., identifying all complement subgraphs when Δ is an edge set and Ω is a family of graphs defined by edge subsets). A *cut* (also known as a *hitting set* [34]) of a set system Ω is a subset of Δ that intersects every $T \in \Omega$. Then, the cut operator $\mathcal{C}(\Omega)$ collects the family of such sets and has seen applications in interdiction games [57]. This cut operator is also closely related to the concepts of *clutters* and *blockers* [18, 21]: a clutter Ω is essentially a set system that only contains minimal elements, while the associated blocker is equivalent to $m(\mathcal{C}(\Omega))$, i.e., the minimal cuts of Ω. In our setting, this cut operator will be used to derive approximations using upper systems. The counterpart of cocut operator is the natural dual of the cut operator and will play a symmetric role in deriving approximations using lower systems. Its definition can be equivalently rewritten as:

$$\begin{aligned}\mathcal{G}(\Omega) &= \{S \subseteq \Delta \mid \forall T \in \Omega,\ (\Delta \setminus S) \cap (\Delta \setminus T) \neq \varnothing\},\\ &= \{S \subseteq \Delta \mid \forall T \in \Omega,\ \Delta \setminus S \not\subseteq T\}.\end{aligned}$$

The first form shows that $\mathcal{G}$ is the cut operator applied to the element-wise complement space $\widehat{\Omega}$; the second reveals that each cocut in $\mathcal{G}(\Omega)$ ensures its complement is not fully contained in any structure $T \in \Omega$. Several elementary algebraic relationships among these operators, termed the *cut–cocut algebra*, are collected in Appendix A to streamline and support later derivations.

2.1 Tightest Monotone Approximations of Set Systems

Building on the previously defined set operators, we present the main results for approximating a set system using upper-systems as follows.

Theorem 1 (Upper Approximation)**.** *For every set system Ω, we have*

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

where $\mathcal{C}(\widehat{\overline{\Omega}})$ and $\mathcal{C}(\widehat{\overline{\uparrow \Omega}})$ are the tightest inner and outer approximations of Ω using upper-systems, respectively. Moreover, equality holds throughout if and only if Ω is upper-closed.

This theorem shows that the two upper-systems, $\mathcal{C}(\widehat{\overline{\Omega}})$ and $\mathcal{C}(\widehat{\overline{\uparrow \Omega}})$, tightly sandwich Ω, providing the best inner and outer approximations among all upper-systems. By symmetry, an analogous result holds for lower approximations as stated below. We omit its proof, as it follows directly from Theorem 1 by applying it to the element-complement space $\widehat{\Omega}$.

Corollary 1 (Lower Approximation)**.** *Given a set system Ω, we have*

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

where $\mathcal{G}(\widehat{\overline{\Omega}})$ and $\mathcal{G}(\widehat{\overline{\downarrow \Omega}})$ are the tightest inner and outer approximations of Ω using lower-systems, respectively. Moreover, equality holds throughout if and only if Ω is lower-closed.

Together, these results yield the tightest inner and outer approximations of any set system Ω using monotone systems. The following corollary extends these constructions to additional cases.

Corollary 2. *Given a set system Ω, the tightest inner approximation by upper-systems 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 inner approximations by lower-systems with equality conditions reversed.

The following example illustrates these results by explicitly computing the inner and outer approximations for Ω_1 and Ω_2 in Figure 1.

Example 1. Consider the set system $\Omega := \Omega_1$ in Figure 1. Computing $\mathcal{C}(\widehat{\overline{\Omega}})$ and $\mathcal{C}(\widehat{\overline{\uparrow \Omega}})$ yields trivial inner and outer approximations $\varnothing$ and $\mathscr{P}(\Delta)$. On the other hand, the lower approximations can be computed as

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

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

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

while the tightest lower approximation counterparts are the trivial set systems. △

2.2 Approximation-Based Reformulations for General BIPs

The preceding approximation results offer two advantages: (i) they yield the tightest possible inner and outer approximations using upper and lower systems; and (ii) they are formulated using cut and cocut operators, which naturally correspond to classical set covering and subtour elimination inequalities, as shown in the following lemma.

Lemma 1. *Given any set system Ω, the vector representations of $\mathcal{C}(\Omega)$ and $\widehat{\mathcal{C}(\Omega)}$ correspond to the following* covering *and* elimination *inequalities, respectively.*

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

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

This lemma leads to the following approximation results for general binary integer programs (BIPs). Let $z(\Pi) \in \mathbb{R} \cup \{\pm\infty\}$ denote the optimal value of an optimization problem Π, where $+\infty$ and $-\infty$ represent infeasibility and unboundedness, respectively.

Theorem 2. *Given a general BIP $\min_{x \in \mathcal{X}} f(x)$, let $\Omega := \{T_x \mid x \in \mathcal{X}\}$. The following reformulations serve as inner/outer approximations of the original problem,*

Upper-Inner Approximation Π_{ui}:

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

Upper-Outer Approximation Π_{uo}:

$$\min_{x \in \{0,1\}^n} \ f(x) \tag{2a}$$
$$s.t. \ \sum_{i \in T} x_i \geq 1,\ \forall T \in m\left(\widehat{\overline{\uparrow \Omega}}\right) \tag{2b}$$

Lower-Inner Approximation Π_{li}:

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

Lower-Outer Approximation Π_{lo}:

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

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, (1) and (2) are both equivalent to the original problem if and only if Ω *is upper-closed; (3) and (4) are both equivalent to the original problem if and only if* Ω *is a lower-closed.*

In other words, every BIP can be approximated from above or below by a pure covering or elimination model. These approximations are exact precisely when the feasible region itself is monotone, explaining why classical covering and elimination formulations succeed in problems like set covering and max-cut, but only approximate more general cases. The following corollary establishes the separation complexity of the above formulations.

Corollary 3. *Suppose* Ω *is monotone with a membership oracle of complexity* $O(\tau(\Omega))$*. For any infeasible binary solution* $x \in \{0,1\}^n$*, the constraint separation complexity for (1)–(4) is of order* $O(\log n \cdot \tau(\Omega))$*.*

Remark 1. Although continuous variables are not the primary focus of this paper, the above results extend naturally to problems of the form $\min_{x \in \mathcal{X} \subseteq \{0,1\}^n, y \in \mathcal{Y} \subseteq \mathbb{R}^m} f(x, y)$, either by applying the approximation results for each fixed feasible y, or by considering the objective function as $g(x) := \min_{y \in \mathcal{Y}} f(x, y)$.

Building on these results, the next example introduces (to the best of the author's knowledge) a new linearization technique for bilinear terms $\langle x, Ry \rangle$ with structured R. In contrast to the standard McCormick approach, which requires auxiliary variables, our reformulation avoids variable lifting altogether and instead leverages covering and elimination inequalities obtained from the monotone system perspective. The goal is not to compete with McCormick in compactness or generality, but to broaden applicability by offering an exact alternative reformulation that aligns naturally with set system structure. In Section 3.2, we further establish facet-defining conditions for these inequalities, and in Section 4.1 extend the reformulation to non-monotone settings.

Example 2 (Linearization of Bilinear Terms)**.** For each supplier $i \in [n]$ and each region $j \in [m]$, $R_{ij} \geq 0$ represents the partnership value (e.g., number of end customers reachable, service compatibility, or market value unlocked) if supplier i supports region j. Then, the following problem seeks to guarantee a partnership value level α while minimizing the total selection cost,

$$\begin{aligned}\min_{x\in\{0,1\}^n, y\in\{0,1\}^m} \quad & \langle c_1, x\rangle + \langle c_2, y\rangle \\ \text{s.t.} \quad & \langle x, Ry\rangle \geq \alpha, \\ & h(x,y) \geq 0,\end{aligned}$$

where the last constraint encodes additional business requirements. A common reformulation method to treat the bilinear term $\langle x, Ry\rangle$ is by introducing auxiliary variables $z_{ij} = x_i y_j$ to initialize the standard linearization step. Since $\langle x, Ry\rangle$ is monotone in (x, y), we can exactly reformulate the problem as follows according to Theorem 2:

$$\begin{aligned}\min_{x\in\{0,1\}^n, y\in\{0,1\}^m} \quad & \langle c_1, x\rangle + \langle c_2, y\rangle \\ \text{s.t.} \quad & \sum_{i\in[n]\setminus I} x_i + \sum_{j\in[m]\setminus J} y_j \geq 1, \quad \forall (I,J) \in M(\overline{\Omega}) \\ & h(x,y) \geq 0,\end{aligned}$$

where the structure set is defined as $\Omega := \{(I,J) \mid \sum_{(i,j)\in I\times J} R_{ij} \geq \alpha\}$, i.e., the index set where the entry-sum of the associated submatrix is above α. Since membership in Ω can be decided in polynomial time, this formulation can be solved via a cut-generation algorithm, using the efficient integer separation procedure guaranteed by Corollary 3.

Suppose the bilinear term appears in the objective function:

$$\begin{aligned}\min_{x\in\{0,1\}^n,\, y\in\{0,1\}^m} \quad & \langle x, Ry\rangle \\ & h(x,y) \geq 0.\end{aligned}$$

Let $z^\star$ be the optimal value and define $\Omega_{z^\star} := \{(I,J) | \sum_{(i,j)\in I\times J} R_{ij} \leq z^\star\}$. Since $R \geq 0$, this system is lower-closed. The problem can then be reformulated as

$$\begin{aligned}\min_{x\in\{0,1\}^n,\, y\in\{0,1\}^m} \quad & \langle R1, x\rangle + \langle R^\intercal 1, y\rangle \\ \text{s.t.} \quad & \sum_{i\in I\cup J} x_i \leq |I \cup J| - 1, \quad \forall (I,J) \in \overline{\Omega_{z^\star}}, \\ & h(x,y) \geq 0.\end{aligned}$$

This reformulation turns the problem into a feasibility problem: every feasible solution must already be optimal because of the elimination constraints, so the objective function is no longer essential for identifying an optimal solution. Although the exact value $z^\star$ is unknown, it can be approached via a cut-generation procedure, similar to those developed for supervalid inequalities in interdiction games [31, 56]. Specifically, we relax the elimination constraints to form a master problem. Solving this master problem at iteration k yields a feasible solution (x_k, y_k) to the original problem with value $\bar{z}_k = \langle x_k, Ry_k \rangle$. Let $\bar{z} := \min_k \bar{z}_k$ be the value of the incumbent, we then assume that the true optimum must be strictly better, set $z^\star := \bar{z} - \epsilon$ for some sufficiently small $\epsilon > 0$, and generate one or more violated elimination constraints to exclude encountered solutions. This process iterates until the master problem becomes infeasible, at which point the best incumbent is guaranteed to be at least ϵ-optimal. Furthermore, if the minimum improvement step in the objective can be bounded away from zero (e.g., when all entries of R are integers), then exact optimality can be certified. In this scheme, the proposed objective function serves primarily as a heuristic guide for producing high-quality candidate solutions in the relaxed master problem. △

This example demonstrates how Theorem 2 can be directly applied to reformulate structured bilinear terms—both in constraints and objectives—without introducing auxiliary variables. The resulting reformulation is complementary rather than competing with existing approaches; for instance, its inequalities can be incorporated into other formulations to further strengthen their relaxations.

More broadly, Theorem 2 yields several key implications that guide the rest of the paper. (i) It provides a unified analytical perspective to identify relevant structures, i.e., the pair $(\Omega, m(\widehat{\overline{\Omega}}))$ for upper-systems and the pair $(\Omega, m(\overline{\Omega}))$ for lower-systems. (ii) It provides a pathway to extend classical results from set covering problems to general BIPs. (iii) It promotes monotone systems as a tool for describing solution spaces, since each such system can be exactly characterized using covering or elimination inequalities, leading to reformulation methods to uncover latent monotone systems. We will explore (i) and (ii) in the next section and develop (iii) in Section 4.

3 Combinatorial and Polyhedral Structures in BIPs

The approximation results developed in Section 2 provide a unified analytical framework for a broad class of binary integer programs (BIPs). This section illustrates its utility from two perspectives: (i) it systematically recovers combinatorial structure pairs that were previously identified in problem-specific contexts; and (ii) it extends classical polyhedral tools—particularly those from the literature on set covering problems—to more general BIPs.

3.1 Recovering Structural Correspondence in Classic Problems

If the objective function is increasing, Theorem 2 implies a strong connection between the structure pairs $(\Omega, \widehat{\overline{\Omega}})$ and $(\Omega, \overline{\Omega})$, corresponding to the respective minimization and maximization problems. In particular, solving one problem requires either covering or eliminating all the structures associated with the other. To make this correspondence precise at the level of convex relaxations, we use

Problem	**Feasible Set Ω**	**Structural Counterpart** $m\left(\widehat{\overline{\uparrow \Omega}}\right)$ for minimization; $m\left(\overline{\downarrow \Omega}\right)$ for maximization
shortest path	s-t paths	s-t vertex cuts / s-t edge cuts
longest path	s-t paths	cycles, claws, subgraphs preventing extension to s-t paths
max-cut	edge cuts	odd simple cycles
min dominating set	dominating sets	closed neighborhoods
min spanning tree	spanning trees	global edge cuts
max spanning tree	spanning trees	simple cycles

Table 2: Structural correspondences computed by the proposed framework. Each problem's feasible set system Ω is paired with its structural counterpart: $m(\widehat{\overline{\uparrow \Omega}})$ for minimization or $m(\overline{\downarrow \Omega})$ for maximization.

$\mathcal{P}_\Omega$ to denote the convex hull of the binary solution space $\mathcal{X}_\Omega$ and define

$$\mathcal{P}_\Omega^{\mathcal{C}} := \left\{ x \in [0,1]^n \,\middle|\, \sum_{i \in T} x_i \geq 1, \forall T \in \Omega \right\},$$
$$\mathcal{P}_\Omega^{\mathcal{G}} := \left\{ x \in [0,1]^n \,\middle|\, \sum_{i \in T} x_i \leq |T| - 1, \forall T \in \Omega \right\}.$$

We use these as LP relaxations of the upper- and lower-type formulations in (1) and (3), with Ω serving as the constraint index set. The following corollary formalizes these relationships in terms of polyhedral solution spaces.

Corollary 4. *Given a monotone system Ω, the following relationships hold,*

$$\textit{When } \Omega \textit{ is upper-closed: } \mathcal{P}_\Omega \subseteq \mathcal{P}^{\mathcal{C}}_{\widehat{\overline{\Omega}}}, \quad \mathcal{P}_{\widehat{\overline{\Omega}}} \subseteq \mathcal{P}^{\mathcal{C}}_\Omega$$
$$\textit{When } \Omega \textit{ is lower-closed: } \mathcal{P}_\Omega \subseteq \mathcal{P}^{\mathcal{G}}_{\overline{\Omega}}, \quad \mathcal{P}_{\overline{\Omega}} \subseteq \mathcal{P}^{\mathcal{G}}_\Omega.$$

Moreover, we have $\min_{x \in \mathcal{P}_\Omega, y \in \mathcal{P}_{\widehat{\overline{\Omega}}}} \langle x, y \rangle \geq 1$ *and* $\max_{x \in \mathcal{P}_\Omega, y \in \mathcal{P}_{\widehat{\overline{\Omega}}}} \langle 1, x + y \rangle - \langle x, y \rangle \leq n - 1$ *for the two cases, respectively.*

Therefore, the relationship between the two polytopes $\mathcal{P}_\Omega$ and $\mathcal{P}_{\widehat{\overline{\Omega}}}$ aligns with the classic theory of blocking polyhedra [24], while the set system pair $(\Omega, \widehat{\overline{\Omega}})$ provides the correspondence structural interpretation in the binary setting.

In what follows, we illustrate these structural correspondences through a series of examples that recover many classical structure pairs from the literature. Although each pair is well known in its problem-specific context, our framework shows that they arise uniformly by computing either $\widehat{\overline{\Omega}}$ or $\overline{\Omega}$, depending on whether Ω is upper- or lower-closed. Throughout the examples, we let $G = (V, E)$ denote a graph with vertex set V and edge set E, and write $N[v]$ for the closed neighborhood of a vertex $v \in V$. Unless stated otherwise, all edge or vertex weights are assumed to be nonnegative. A summary of these structural relationships is provided in Table 2.

Example 3 (Shortest Path)**.** In the shortest path problem with nonnegative edge lengths, the set system Ω consists of all s-t paths. Since this is a minimization problem with an increasing objective function, the feasible region can be extended to the upper system $\uparrow \Omega$, which includes all s-t connected subgraphs. A direct computation shows that the associated structures in $\widehat{\uparrow \Omega}$ correspond to subgraphs whose removal disconnects s from t. Depending on whether one focuses on edges or vertices, the minimal structures $m(\widehat{\uparrow \Omega})$ yield either the set of s-t edge cuts or vertex cuts. This correspondence recovers the structural aspect of the classic max-flow min-cut theorem [19] and the vertex version of Menger's theorem [44], respectively. △

Example 4 (Longest Path)**.** In the longest path problem with nonnegative edge lengths, the set system $\downarrow \Omega$ consists of all subgraphs that can be extended to an s-t path. The corresponding structure set $\overline{\downarrow \Omega}$ includes subgraphs that cannot be extended to such a path. While this set lacks a simple and complete characterization, several sufficient conditions are readily verifiable. In particular, the following subgraphs belong to $\overline{\downarrow \Omega}$:

- Subgraphs containing claws;
- Subgraphs containing cycles;
- Subgraphs with more than one edge incident to either s or t;
- Subgraphs in directed graphs with vertices having more than one in-degree or out-degree.

The first two types of subgraphs are commonly investigated in problems related to longest paths [13, 37]. Although these structures are not sufficient to fully describe $\overline{\downarrow \Omega}$, they can be heuristically generated or iteratively separated to strengthen relaxations when solving the associated binary integer program. For example, various forms of cycle elimination constraints have been used in [42]. △

Example 5 (Max-Cut)**.** With nonnegative edge weights, the lower system $\downarrow \Omega$ consists of edge sets that can be extended to an edge cut—that is, all subgraphs that are bipartite (any bipartite subgraph admits a 2-coloring, which extends to a partition of V, making it a subset of a cut). Then, the minimal structures in $\overline{\downarrow \Omega}$ are precisely the odd cycles, as these are the smallest subgraphs that violate bipartiteness. Therefore, solving the max-cut problem is equivalent to eliminating all odd cycles. This structural perspective aligns with classical results on the cut polytope and the max-cut problem [6, 7, 26]. △

Example 6 (Minimum Dominating Set)**.** A dominating set is a vertex subset $T \subseteq V$ such that every vertex in V is either in T or adjacent to some vertex in T. By definition, the family of dominating sets Ω is upper-closed. The associated structural system $m(\widehat{\Omega})$ can be derived as:

- $\overline{\Omega}$ consists of non-dominating sets, i.e., subsets $T \subseteq V$ for which there exists a vertex $v \in V \setminus T$ that has no neighbors in T.

- $\widehat{\widehat{\Omega}}$ contains all subsets T that include some vertex $v \in T$ such that the closed neighborhood $N[v] \subseteq T$.
- The minimal such subsets are exactly the closed neighborhoods themselves: $m(\widehat{\widehat{\Omega}}) = \{N[v]\}_{v \in V}$.

Thus, covering all closed neighborhoods is necessary and sufficient for vertex domination, a structural correspondence that has been previously explored in [28]. △

Example 7 (Spanning Trees)**.** In the spanning tree minimization problem, the upper system $\uparrow \Omega$ consists of all connected subgraphs. The associated structural set $m(\widehat{\widehat{\Omega}})$ then corresponds to all edge cuts—minimal sets of edges whose removal disconnects the graph. In contrast, when maximizing over spanning trees, we consider the lower system $\downarrow \Omega$, which consists of all forests (acyclic subgraphs). The corresponding structure set $m(\overline{\Omega})$ contains all simple cycles, as these are the minimal subgraphs that violate acyclicity. The first correspondence underlies the design of efficient min-cut algorithms based on spanning tree packings [33], while the second reflects the classical matroidal duality between bases (maximal independent sets) and circuits (minimal dependent sets) [48]. △

Similar analyses can be applied to other set systems, such as vertex covers, independent sets, and matchings, by computing $\widehat{\widehat{\Omega}}$ or $\overline{\Omega}$, depending on whether Ω is upper- or lower-closed. These dual systems reveal meaningful structural counterparts and consistently recover well-known combinatorial patterns observed across classical problems.

3.2 Generalizing Polyhedral Analysis for BIPs

Another unifying perspective offered by Theorem 2 is that if either the solution space Ω or the objective function f is monotone, the binary optimization problem $\min_{x \in \mathcal{X}_\Omega} f(x)$ can be reformulated exactly as a set covering problem (1) or an elimination problem (3). This reformulation enables direct application of the rich body of polyhedral results developed for set covering problems, including facet characterizations and lifting techniques, thereby systematically extending these tools to the analysis of general BIPs. To unify the facet analysis of covering- and elimination-type problems, we first introduce the following facet-preserving result.

Theorem 3. *Given a binary solution space $\mathcal{X} := \{x \in \{0,1\}^n \mid g(x) \leq 0\}$ and an index set $I \subseteq [n]$, define the flipping map $\theta_I : \{0,1\}^n \to \{0,1\}^n$ by*

$$(\theta_I(x))_i := \begin{cases} 1 - x_i & \text{if } i \in I, \\ x_i & \text{otherwise.} \end{cases}$$

Let $\mathcal{X}_{\theta_I} := \{x \in \{0,1\}^n \mid g(\theta_I(x)) \leq 0\}$. Then, θ_I induces a bijective affine map between $\operatorname{conv}(\mathcal{X})$ and $\operatorname{conv}(\mathcal{X}_{\theta_I})$. In particular, an inequality $\langle a, x \rangle \geq b$ is valid (or facet-defining) for $\operatorname{conv}(\mathcal{X})$ if and only if $\langle a, \theta_I(x) \rangle \geq b$ is valid (or facet-defining) for $\operatorname{conv}(\mathcal{X}_{\theta_I})$.

Since the elimination problem (3) can be reformulated as a covering problem (1) via the flipping map $\theta_{[n]}$, its facet analysis can be unified with that of the covering problem. This equivalence allows general techniques from the set covering literature—including facet-defining conditions [4, 5, 51], lifting-based constraint strengthening [47, 56], and supervalid inequalities derived from solution bipartition [57]—to be extended to a broader class of binary integer programs. While a comprehensive treatment is beyond the scope of this paper, we illustrate this unifying potential by presenting a facet-defining criterion for general BIPs adapted from the existing literature [5, 56], which requires the following definition.

Definition 3 (Quasi-Feasibility)**.** Let $T \in \overline{\Omega}$ be an infeasible (non-solution) structure, we say T is *quasi-feasible* if, for every element $a \in T$, there exists an element $a' \in \Delta \setminus T$ such that replacing a with a' yields a feasible set; that is, $T' := T \setminus \{a\} \cup \{a'\} \in \Omega$.

This concept identifies infeasible sets that are locally close to feasibility, providing general insight into the facet-defining conditions of covering-type constraints.

Corollary 5. *Suppose Ω is an upper system and $|T| \geq 2$ for every $T \in m(\widehat{\overline{\Omega}})$, the covering inequality (1b) is facet-defining if and only if $\Delta \setminus T \in M(\overline{\Omega})$ is quasi-feasible.*

This corollary illustrates how the proposed framework could be used to bridge the rich theory of set covering with the analysis of a broad class of BIPs, enabling established tools to be systematically extended to new problem classes and their associated solution structures.

Example 8 (Linearization of Bilinear Terms (Continued))**.** Using Corollary 5, we can derive the facet conditions for the following covering constraints from Example 2,

$$\sum_{i \in [n] \setminus I} x_i + \sum_{j \in [m] \setminus J} y_j \geq 1, \quad \forall (I, J) \in M(\overline{\Omega}),$$

where $\Omega := \{(I, J) \mid \sum_{(i,j) \in I \times J} R_{ij} \geq \alpha\}$. Suppose every submatrix obtained by removing only one row or one column from R has its entry-sum satisfies the required level α, the above linear constraint is facet-defining if and only if the following two conditions are both satisfied,

- For every row $i \in I$, we can find either a substitution row i' or a substitution column j' so that the new submatrix $R_{I \setminus \{i\} \cup \{i'\}, J}$ or $R_{I \setminus \{i\}, J \cup \{j'\}}$ meets the required level α;
- For every column $j \in J$, we can find either a substitution row i' or a substitution column j' so that the new submatrix $R_{I \cup \{i'\}, J \setminus \{j\}}$ or $R_{I, J \setminus \{j\} \cup \{j'\}}$ meets the required level α.

This is a direct interpretation of quasi-feasibility in this specific case. △

By recovering well-known combinatorial correspondences and extending polyhedral tools, this section unifies a broad class of existing results. The next section goes further by introducing reformulation methods based on the identification of latent monotone structures, such as bimonotone and interval systems.

4 Reformulations via Latent Monotone Systems

The exactness of reformulations (1)–(3) requires monotonicity of either the objective or feasible region. This section extends their scope by uncovering latent monotone structures within general BIPs, through two complementary strategies: (i) bimonotone cuts; and (ii) interval system decomposition. We also illustrate these strategies through three examples that reveal latent monotone structure in different settings: (1) signed min-cut with mixed edge weights, where bimonotone cuts provide an exact linear reformulation; (2) piecewise monotone objectives, such as those learned from tree-based regression models; and (3) disjunctive multi-criteria optimization, where interval decompositions naturally apply. These examples demonstrate how latent monotone systems translate into concrete reformulation tools.

4.1 Bimonotone Cuts

The first source of latent monotone structure comes from functions that are monotone with opposite signs on a bipartition of the variables. We need the following two definitions, where a bipartition is said to be trivial if one of the two parts is empty.

Definition 4 (Bimonotone Function)**.** A function $g : \{0,1\}^n \to \mathbb{R}$ is called bimonotone if there exist some (possibly trivial) bipartition (I, J) of $[n]$ such that for every $x \in \{0,1\}^I$ and every $y \in \{0,1\}^J$, the functions $g(x,\cdot) : \{0,1\}^J \to \mathbb{R}$ and $g(\cdot,y) : \{0,1\}^I \to \mathbb{R}$ are increasing and decreasing, respectively.

Definition 5 (Bimonotone Closure)**.** Given a set system Ω on ground set $[n]$ and a (possibly trivial) bipartition (I, J) of $[n]$, for every $T \in \Omega$ we write $T_I := T \cap I$ and $T_J := T \cap J$. The *bimonotone closure* of Ω with respect to (I, J) is defined as

$$\uparrow_I\downarrow_J \Omega := \{\, T \subseteq [n] \mid T_I \supseteq T'_I \text{ and } T_J \subseteq T'_J \text{ for some } T' \in \Omega \,\}.$$

For every bimonotone system Ω, we define its *extremal elements* as the structures $(T_I, T_J) \in \uparrow_I\downarrow_J \Omega$ such that removing any element from T_I or adding any element to T_J produces a structure not belonging to $\uparrow_I\downarrow_J \Omega$. We denote this set of extremal elements by $\mathcal{E}(\uparrow_I\downarrow_J \Omega)$.

While bimonotonicity can be reduced to monotonicity through the flipping map in Theorem 3, this connection has rarely been exploited to derive monotone cut variants. Our goal is to translate this observation into a concrete reformulation tool, thereby extending the reach of exact monotone reformulations to a broader class of BIPs by incorporating the following wider family of functions.

Proposition 1. *The following functions are bimonotone:*

- *Linear (modular) functions.*
- *Bilinear functions $\langle x, Ry\rangle$ for some block-diagonal matrix $R \in \mathbb{R}^{I\times J}$ where there exist (possibly trivial) index bipartitions $I = I_1 + I_2$ and $J = J_1 + J_2$ such that the diagonal blocks satisfy $R_{I_1J_1} \geq 0$ and $R_{I_2J_2} \leq 0$ (entrywise).*

- *Submodular functions f where for every $i \in \Delta$, either*

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

- *The supermodular counterpart.*

Once such a bimonotone objective function is identified in a given BIP, we obtain an exact reformulation method and facet-defining conditions according to the following theorem.

Theorem 4. *Given a BIP $\min_{x=(x_I,x_J)\in\mathcal{X}_\Omega} f(x_I, x_J)$ with a bimonotone objective f that is increasing and decreasing in x_I and x_J, respectively. Then, an equivalent reformulation is:*

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

$$\text{s.t.} \sum_{i\in T_I} x_i + \sum_{j\in J\setminus T_J} (1 - x_j) \ge 1, \ \forall (T_I, T_J) \in \mathcal{E}\left(\widehat{\uparrow_I\downarrow_J \Omega}\right). \tag{5b}$$

Moreover, suppose $|T_I \cup (J \setminus T_J)| \ge 2$ for every $(T_I, T_J) \in \widehat{\uparrow_I\downarrow_J \Omega}$, the above inequality is facet-defining if and only if the following two conditions are both satisfied:

- *For every $i \in I \setminus T_I$, there exists some $i' \in T_I$ such that $((I \setminus T_I) \setminus \{i\} \cup \{i'\}, T_J) \in \uparrow_I\downarrow_J \Omega$;*
- *For every $j \in T_J$, there exists some $j' \in J \setminus T_J$ such that $(I \setminus T_I, T_J \setminus \{j\} \cup \{j'\}) \in \uparrow_I\downarrow_J \Omega$.*

By interpreting bimonotone functions through the flipping map, this reformulation follows directly from Theorem 2, while the facet condition is an application of the general criterion in Corollary 5. What is new here is that, via the identification of bimonotone functions, these tools can now be applied to less obvious cases that would otherwise fall outside the classical monotone framework. Similarly, if bimonotone functions are used for defining constraints, the same reformulation can be applied as shown in the following corollary.

Corollary 6. *Given a constraint of a BIP defined by $g(x_I, x_J) \ge 0$ for some bimonotone function g that is increasing in x_I and decreasing in x_J, then all the structures satisfy this constraint form a set system Ω that is bimonotone, i.e., $\Omega = \uparrow_I\downarrow_J \Omega$. In particular, the associated constraint set (5b) provides a linear representation of $g(x_I, x_J) \ge 0$.*

Leveraging bimonotone functions, these results can assist to analyze the structures tied to (5b) and separate valid inequalities or even facets to strengthen the associated BIP. We demonstrate this with the following example.

Example 9 (Min-Cut with Signed Weights)**.** Consider a graph $G = (V, E)$ with signed edge weights $\{w_e\}_{e\in E}$. The signed min-cut problem seeks an edge cut of minimum total weight. When all weights are nonnegative, this reduces to the classical min-cut, while its complement corresponds to max-cut. In contrast, mixed signs generally destroy the PSD matrix structure, so the problem no longer admits the convex QP formulation available in the nonnegative case.

Algorithm 1 Linear-Time Membership Oracle of $\uparrow_I\downarrow_J \Omega$ for Signed Min-Cut

input: an edge set T, an edge bipartition (I, J)
output: whether $T \in\uparrow_I\downarrow_J \Omega$
contract vertices in $I \setminus T_I$ to form vertex classes V'
construct graph $H = (V', E')$ where $(v_1, v_2) \in E'$ if some $e \in T_J$ connects v_1 and v_2
return whether H is bipartite

By Theorem 4, the edge set can be partitioned as $E = (I := E_+, J := E_-)$, where E_+ collects all nonnegative edges. The objective function is then bimonotone with respect to this bipartition, and formulation (5) yields an exact linear representation of the signed min-cut problem.

In this setting, $\uparrow_I\downarrow_J \Omega$ consists of edge subsets (T_I, T_J) such that $T_I \supseteq T'_I$ and $T_J \subseteq T'_J$ for some cut (T'_I, T'_J). Equivalently, $(T_I, T_J) \in\uparrow_I\downarrow_J \Omega$ if and only if the vertices incident to $I \setminus T_I$ can be contracted into a single color class, and every edge in T_J connects distinct color classes. Algorithm 1 implements this criterion: it contracts $I \setminus T_I$, builds the residual graph, and checks bipartiteness in $O(|V| + |E|)$ time.

Thus, the bimonotone reformulation provides both an exact linear model and a polynomial-time membership oracle. By Corollary 3, this enables efficient integer separation for the inequalities in (5b), enabling the associated cut-generation implementation. △

4.2 Interval System Decomposition

In addition to bimonotone functions, latent monotone structure can also be revealed through set systems that exhibit the interval property, as characterized below.

Definition 6 (Interval System)**.** A set system Ω is said to have the interval property, or to be an interval system, if for all $T, T' \in \Omega$ and any T'' with $T \subseteq T'' \subseteq T'$, it follows that $T'' \in \Omega$.

This property admits a concise characterization: interval systems can be represented as the intersection of their upward and downward closures.

Proposition 2. *For any set system Ω, $\uparrow \Omega \cap \downarrow \Omega$ is the smallest interval system that contains Ω. Moreover, Ω is interval if and only if $\Omega = \uparrow \Omega \cap \downarrow \Omega$.*

Therefore, every identified interval system Ω can be represented as follows

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

The next proposition establishes that, in theory, any binary solution space can be expressed as a union of interval systems, and therefore admits a representation entirely in terms of covering and elimination inequalities.

Proposition 3. *Every set system Ω adopts a decomposition $\Omega = \bigcup_{k \in K} \Omega_k$ for some interval set systems $\{\Omega_k\}_{k \in K}$, which leads to the following exact reformulation of the associated problem*

$\min_{x\in\mathcal{X}_\Omega} f(x)$

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

$$s.t. \; \sum_{i\in T} x_i \geq z_k, \qquad \forall k\in K, \forall T \in m\left(\widehat{\overline{\uparrow \Omega_k}}\right), \tag{6b}$$

$$\sum_{i\in T} x_i \leq |T| - z_k, \; \forall k\in K, \forall T \in m\left(\overline{\downarrow \Omega_k}\right), \tag{6c}$$

$$\sum_{k\in K} z_k = 1, \tag{6d}$$

where z_k represents the set system to which the optimal solution belongs.

Although such a decomposition always exists (even via trivial singleton intervals as used in the proof), its practical value arises when the number of components is small or structured, a situation that naturally occurs in disjunctive-type problems [2]. The following examples illustrate how nontrivial interval decompositions naturally emerge in structured settings, enabling exact reformulations that extend well beyond the trivial singleton case.

Example 10 (Piecewise Monotone Objectives)**.** Suppose the objective function f is piecewise monotone. Specifically, let $\{\mathcal{X}_k\}_{k\in K}$ be a partition of the hypercube $[0,1]^n$, and assume that on each region $\mathcal{X}_k$ the function f takes the affine form $f(x) = \langle c^k, x\rangle + c_0^k$, where c^k is either nonnegative or nonpositive. Define K_+ (resp. K_-) as the index set of regions where f is increasing (resp. decreasing). Then the problem $\min_{x\in\mathcal{X}_\Omega} f(x)$ admits the following exact reformulation:

$$\begin{aligned} \min_{x\in\{0,1\}^n, z\in\{0,1\}^{|K|}} \; & \eta \\ \text{s.t. } & \eta \geq \langle c^k, x\rangle + c_0^k - M(1-z_k), && \forall k\in K, \\ & \sum_{i\in T} x_i \geq z_k, && \forall k \in K_+, \forall T\in m\left(\widehat{\overline{\uparrow (\Omega\cap\Omega_{\mathcal{X}_k})}}\right), \\ & \sum_{i\in T} x_i \leq |T| - z_k, && \forall k\in K_-, \forall T\in m\left(\overline{\downarrow (\Omega\cap\Omega_{\mathcal{X}_k})}\right), \\ & \sum_{k\in K} z_k = 1, \end{aligned}$$

where $M > 0$ is a sufficiently large constant. Here, the binary variable z_k encodes the active region $\mathcal{X}_k$ for the solution x. Then, for $k\in K_+$, f is increasing over $\Omega\cap\Omega_{\mathcal{X}_k}$ and the covering inequalities apply; for $k\in K_-$, f is decreasing and the elimination inequalities apply. This setting arises naturally when f is learned from data. For example, tree-based regression models yield a piecewise-affine predictor, where each leaf corresponds to a region $\mathcal{X}_k$ and provides coefficients (c^k, c_0^k) [35, 39]. In a network design context, $f(x)$ might represent the latency of a connected subgraph Ω learned from simulation or historical data [22, 41]. Once such a surrogate objective is available, the above reformulation provides an exact optimization model under the learned piecewise monotone structure. The same approach applies to reformulate any piecewise monotone constraint of the form $g(x)\geq 0$. △

Example 11 (Disjunctive Multi-Criteria Optimization)**.** Consider a monotone binary space $\mathcal{X}_{\Omega_0}$. For each $k \in [K]$, let $l_k \le g_k(x) \le u_k$ define the k-th disjunctive criterion (e.g., a prescribed range for cost, profit, or utility) under some monotone criterion function $g_k(x)$. The resulting set system can be written as

$$\Omega \;=\; \{\, T \in \Omega_0 \;\mid\; l_k \le g_k(x_T) \le u_k \text{ for some } k \in [K]\,\}.$$

This system naturally decomposes into K interval subsystems,

$$\Omega_k \;=\; \{\, T \in \Omega_0 \;\mid\; l_k \le g_k(x_T) \le u_k \,\}, \quad k \in [K].$$

Formulation (6) can then be applied, either to represent the entire feasible space Ω or to selectively separate inequalities that strengthen the relaxation. △

Both examples extend naturally to the bimonotone setting (i.e., piecewise bimonotone functions and bimonotone g_k's in the respective cases), where the inequalities (6b)-(6c) are replaced by the bimonotone cuts in (5b). Taken together, these results show that latent monotone systems—whether bimonotone or interval—provide a unified reformulation framework for BIPs, broadening the scope of monotone reformulations and connecting them to disjunctive and piecewise structures.

5 Case Study: Network Site Selection under Uncertainty

This case study illustrates the two central aspects of our framework. First, it utilizes the interval system reformulation: the feasible space form an interval system, so the reformulation via covering and elimination inequalities is exact. Second, the study highlights flexibility: by choosing different monotone subsystems for reformulation, we will obtain four hybrid implementations, each with distinct computational trade-offs.

Although our numerical study focused on network site selection, the same reformulation strategies extend naturally to other domains. For example, monotone and bimonotone subsystems occur often in survivability, scheduling, routing, and interdiction problems. In each case, identifying the relevant subsystems enables hybrid implementations of covering and elimination cuts similar to those explored here.

5.1 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 are 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.

$$\max_{x \in \{0,1\}^V} \sum_{i \in V} r_i x_i \tag{7a}$$

$$\text{s.t. } \bar{\mathbb{P}}_j \left(\sum_{i \in \delta[j]} a_{ij} x_i \geq b_j \right) \geq 1 - \epsilon_j, \ \forall j \in V \tag{7b}$$

$$\sum_{i \in C} x_i \leq 1, \qquad \forall C = \{i, j\} \in E. \tag{7c}$$

The objective (7a) is to maximize the net benefit. (7b) contains the chance constraints to impose demand satisfaction requirements with $\delta[j]$ denoting the closed neighborhood of vertex j (i.e., $\{j\} \cup \{i : \{i, j\} \in E\}$), and (7c) ensures that the resulting solution is an independent set. Moreover, (7c) can be further enhanced using clique elimination constraints.

From the perspective of Section 2, constraints (7b) and (7c) define two monotone subsystems: $\Omega_{\mathcal{X}_1}$ is an upper system (chance constraints with nonnegative coefficients), while $\Omega_{\mathcal{X}_2}$ is a lower system (independent set constraints). Hence, the feasible region $\Omega = \Omega_{\mathcal{X}_1} \cap \Omega_{\mathcal{X}_2}$ forms an interval system. This classification yields multiple reformulation paths: covering inequalities can be applied to $\Omega_{\mathcal{X}_1}$, elimination inequalities to $\Omega_{\mathcal{X}_2}$, or both simultaneously. In particular, the chance-constraint set (7b) correspond to an upper system, so Theorem 2 reformulates them into the following inequalities.

$$\sum_{i \in T} x_i \geq 1, \qquad \forall T \in m\left(\widehat{\overline{\Omega_{\mathcal{X}_1}}}\right). \tag{8}$$

A direct computation shows that each T in this index set corresponds to a minimal collection of sites whose complement yields a satisfaction probability strictly less than $1 - \epsilon_j$. Theorem 2 guarantees that this reformulation is exact, while Corollary 5 characterizes the facet-defining inequalities. Moreover, classical strengthening techniques for set covering—such as lifting and supervalid inequalities [47, 51, 56, 57]—become immediately applicable to enhance the original chance constraints. This illustrates how the monotone-system perspective not only enables flexible reformulations but also unifies their analysis and strengthening within a broader polyhedral toolkit.

5.2 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, each of which corresponds to a different way of exploiting the two monotone systems $\Omega_{\mathcal{X}_1}$ and $\Omega_{\mathcal{X}_2}$.

- NoCut: Baseline finite scenario approximation (FSA) algorithm; no monotone cuts used.
- ClqCut: Exploits lower system $\Omega_{\mathcal{X}_2}$; adds clique-based elimination cuts.
- SatCut: Exploits upper system $\Omega_{\mathcal{X}_1}$; generates covering cuts (8).
- AllCut: Uses both, consistent with interval-system structure $\Omega_{\mathcal{X}_1} \cap \Omega_{\mathcal{X}_2}$.

Thus, clique cuts correspond to classical polyhedral strengthening for independent set problems, while satisfaction cuts mirror Benders/LBBD-style no-good cuts in chance-constrained settings. Our framework shows that both arise naturally as monotone inequalities.

In NoCut and SatCut, we reformulate (7b) into the following constraint set using standard finite scenario sampling method [55],

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

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

where a_{ij}^k's and b_j^k's are sampled from the nominal distribution $\bar{\mathbb{P}}_j$. In ClqCut, we generate cuts for (7c) on-the-fly by identifying up to three violated *maximal cliques* (with $|C| \geq 2$) in the induced subgraph $G[T]$ at an incumbent solution 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 (7b). For the rejection case, the proposed constraint (8) will be added to the master according to (6).

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 Beta(2, 2) if $\{i,j\} \in E$, and $a_{ij} = 0$ otherwise. The demands b_j are uniformly distributed on $[0, 0.1]$. The benefit coefficients r_i are integers drawn uniformly from $\{-20, \ldots, 20\}$.

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 3.

In terms of average runtime, SatCut and AllCut significantly outperform the other two implementations, with SatCut having a slight edge in efficiency. Implementing clique cuts for (7c) improves runtime relative to NoCut, though not to the extent achieved by SatCut and AllCut. From the "Num. of Cuts" column, SatCut attains its performance with the fewest cuts, indicating strong inequalities that effectively tighten the relaxation. 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 (7b) with fixed constraints in (7c) demonstrates

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 3: 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. This confirms that exploiting the upper system $\Omega_{\mathcal{X}_1}$ through satisfaction cuts is most effective in this setting.

the best performance. We also note that the $|V|$ subproblems for generating (8) can be further parallelized due to the constraint-wise independence in (7b), which could further improve the efficiency of the SatCut and AllCut implementations.

5.3 Distributionally Robust Chance Constraints

Having compared these reformulations under nominal distributions, we next show how the same framework extends to distributionally robust chance constraints. In practice, the nominal distribu-

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 4: 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.

tion $\bar{\mathbb{P}}_j$ 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 (7b) is often used.

$$\sup_{\mathbb{P}_j \in \mathfrak{P}(\bar{\mathbb{P}}_j)} \mathbb{P}_j \left(\sum_{i \in \delta[j]} a_{ij} x_i < b_j \right) = \sup_{\mathbb{P}_j \in \mathfrak{P}(\bar{\mathbb{P}}_j)} \mathbb{E}_{\mathbb{P}_j} \left[\mathbb{I}_{\Xi_j(x)} \right] \leq \epsilon_j, \ \forall j \in V, \tag{9}$$

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 $\bar{\mathbb{P}}_j$.

Although the DRO chance constraint is more complex, the associated feasible structures in $\Omega_{\mathcal{X}_1}$ still form an upper system under our monotone-system perspective, regardless of the specific ambiguity set $\mathfrak{P}(\bar{\mathbb{P}}_j)$. Consequently, the same family of covering inequalities as in (8) remains valid, with separation handled by a tailored oracle. This shows that the proposed framework extends naturally to distributionally robust settings.

For illustration, 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 $\{(a_j^k, b_j^k)\}_{k\in[K]}$, each problem $\sup_{\mathbb{P}_j \in \mathfrak{P}(\bar{\mathbb{P}}_j)} \mathbb{E}_{\mathbb{P}_j}\left[\mathbb{I}_{\Xi_j(x)}\right]$ can be exactly reformulated to the following dual problem [45].

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

Note that the set indicator function is not piecewise concave, thus we cannot use the reformulation method introduced by [45]. Instead, we can use the following finite scenario approximation method assuming Ξ is fully supported on the samples.

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

In the DRO-SatCut implementation, we use this linear programming as the subproblem for each $j \in V$ to separate (8). For comparison, the DRO-NoCut algorithm incorporates the above formulation as constraints to obtain the following MIP,

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

Using the same experiment setting as before with the radius $\eta = 0.1$, we present the experiment results in Table 4.

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 covering constraints (8).

Overall, this case study demonstrates two central features of our framework: (i) by classifying constraints into monotone systems, we obtain a flexible reformulation toolkit; and (ii) multiple monotone systems can be identified in the same problem, recovering known techniques (clique cuts, Benders/LBBD no-good cuts) and enabling hybrid implementation (AllCut). The empirical results confirm that this structural perspective translates into flexible reformulation methods and concrete performance benefits.

6 Conclusion

This paper develops a unified framework for set system approximation in binary integer programs (BIPs), placing covering and elimination inequalities at the center of solution space characterization, rather than as problem-specific or infeasibility-driven constructs. The framework addresses the three motivating inquiries posed in the introduction. For the first inquiry, we established that covering and elimination inequalities correspond exactly to monotone inner and outer approximations, identifying when they yield valid and exact characterizations of binary solution spaces. For the second inquiry, we showed how this perspective systematically recovers classical structural correspondences and extends polyhedral tools from set covering to arbitrary BIPs, including facet-defining conditions and lifting techniques. For the last one, we proposed new reformulation tools for nonlinear and latent monotone systems, including bilinear linearization without auxiliary variables, bimonotone cuts, and interval system decompositions, that broaden the reach of monotone-based reformulations well beyond classical settings.

A case study on distributionally robust network site selection demonstrated how different combinations of monotone subsystems naturally yield multiple hybrid implementations, confirming the flexibility of the framework. Beyond this case study, the framework applies broadly to binary optimization problems with latent monotone structure, offering both theoretical insights and practical reformulation strategies.

Several directions for future work remain. From a theoretical standpoint, exploring other set operators and their associated inequalities may lead to new reformulation strategies. From a computational perspective, integrating parallelization and learning methods into the reformulation of piecewise monotone problems (Example 10) may further enhance scalability. Finally, extending these ideas to mixed-integer nonlinear programs represents a promising avenue for broadening the scope of monotone-based reformulations. In summary, this framework expands the toolkit of valid inequalities and reformulations available in BIPs, providing both theoretical insights and computational benefits.

References

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

[2] Egon Balas. Disjunctive programming. *Annals of discrete mathematics*, 5:3–51, 1979.

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

[4] Egon Balas and Shu Ming Ng. On the set covering polytope: Ii. lifting the facets with coefficients in {0, 1, 2}. *Mathematical Programming*, 45(1):1–20, 1989.

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

[6] Francisco Barahona. The max-cut problem on graphs not contractible to k5. *Operations Research Letters*, 2(3):107–111, 1983.

[7] Francisco Barahona and Ali Ridha Mahjoub. On the cut polytope. *Mathematical programming*, 36(2):157–173, 1986.

[8] Francisco Barahona, Martin Grötschel, and Ali Ridha Mahjoub. Facets of the bipartite subgraph polytope. *Mathematics of Operations Research*, 10(2):340–358, 1985.

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

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

[11] Kyle EC Booth, Tony T Tran, and J Christopher 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.

[12] Quentin Botton, Bernard Fortz, Luis Gouveia, and Michael Poss. Benders decomposition for the hop-constrained survivable network design problem. *INFORMS journal on computing*, 25 (1):13–26, 2013.

[13] Hajo Broersma, Fedor V Fomin, Pim van't Hof, and Daniël Paulusma. Exact algorithms for finding longest cycles in claw-free graphs. *Algorithmica*, 65(1):129–145, 2013.

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

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

[16] Alistair R Clark, Reinaldo Morabito Neto, and Eli AV 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.

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

[18] Gérard Cornuéjols. *Combinatorial optimization: Packing and covering*. SIAM, 2001.

[19] George Dantzig and Delbert Ray Fulkerson. On the max flow min cut theorem of networks. *Linear inequalities and related systems*, 38:225–231, 2003.

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

[21] Jack Edmonds and Delbert Ray Fulkerson. Bottleneck extrema. *Journal of Combinatorial Theory*, 8(3):299–306, 1970.

[22] Amr Elfar, Alireza Talebpour, and Hani S Mahmassani. Machine learning approach to short-term traffic congestion prediction in a connected environment. *Transportation Research Record*, 2672(45):185–195, 2018.

[23] Lester R Ford Jr and Delbert R Fulkerson. Maximal flow through a network. *Canadian journal of Mathematics*, 8:399–404, 1956.

[24] Delbert R Fulkerson. Blocking polyhedra. Technical report, 1968.

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

[26] Martin Grötschel and George L Nemhauser. A polynomial algorithm for the max-cut problem on graphs without long odd cycles. *Mathematical Programming*, 29(1):28–40, 1984.

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

[28] Teresa W Haynes, Stephen Hedetniemi, and Peter Slater. *Fundamentals of domination in graphs*. CRC press, 2013.

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

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

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

[32] Imdat Kara, Gilbert Laporte, and Tolga 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.

[33] David R Karger. Minimum cuts in near-linear time. *Journal of the ACM (JACM)*, 47(1): 46–76, 2000.

[34] Richard M Karp. Reducibility among combinatorial problems. In *50 Years of Integer Programming 1958-2008: from the Early Years to the State-of-the-Art*, pages 219–241. Springer, 2009.

[35] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[36] Gilbert Laporte, Yves Nobert, and Danielle Arpin. An exact algorithm for solving a capacitated location-routing problem. *Annals of Operations Research*, 6:291–310, 1986.

[37] MingChu Li. Longest cycles in almost claw-free graphs. *Graphs and Combinatorics*, 16(2): 177–198, 2000.

[38] Yantong Li, Jean-François Côté, Leandro Callegari-Coelho, and Peng 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.

[39] Wei-Yin Loh. Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 1(1):14–23, 2011.

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

[41] Xiaolei Ma, Haiyang Yu, Yunpeng Wang, and Yinhai Wang. Large-scale transportation network congestion evolution prediction using deep learning theory. *PloS one*, 10(3):e0119044, 2015.

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

[43] Garth P McCormick. Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems. *Mathematical programming*, 10(1):147–175, 1976.

[44] Karl Menger. Zur allgemeinen kurventheorie. *Fundamenta Mathematicae*, 10(1):96–115, 1927.

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

[46] Imen Ben Mohamed, Walid Klibi, Ruslan Sadykov, Halil Şen, and François Vanderbeck. The two-echelon stochastic multi-period capacitated location-routing problem. *European journal of operational research*, 306(2):645–667, 2023.

[47] Paolo Nobili and Antonio Sassano. Facets and lifting procedures for the set covering polytope. *Mathematical Programming*, 45(1):111–137, 1989.

[48] James G Oxley. *Matroid theory*, volume 3. Oxford University Press, USA, 2006.

[49] Youngsoo Park, Peter Nielsen, and Ilkyeong Moon. Unmanned aerial vehicle set covering problem considering fixed-radius coverage constraint. *Computers & Operations Research*, 119: 104936, 2020.

[50] Ragheb Rahmaniani, Teodor Gabriel Crainic, Michel Gendreau, and Walter Rei. The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259 (3):801–817, 2017.

[51] Miguel Sánchez-García, Marıa Inés Sobrón, and Begoña Vitoriano. On the set covering polytope: Facets with coefficients in {0, 1, 2, 3}. *Annals of Operations Research*, 81(0):343–356, 1998.

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

[53] J Cole Smith and Yongjia Song. A survey of network interdiction models and algorithms. *European Journal of Operational Research*, 283(3):797–811, 2020.

[54] Theunis HC Smith, V Srinivasan, and GL 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.

[55] Yongjia Song, James R Luedtke, and Simge Küçükyavuz. Chance-constrained binary packing problems. *INFORMS Journal on Computing*, 26(4):735–747, 2014.

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

[57] Ningji Wei and Jose L Walteros. On supervalid inequalities for binary interdiction games. *Mathematical Programming*, pages 1–42, 2024.

[58] Ningji Wei, Jose L Walteros, and Foad Mahdavi Pajouh. Integer programming formulations for minimum spanning tree interdiction. *INFORMS Journal on Computing*, 33(4):1461–1480, 2021.

[59] Yiming Yao, Thomas Edmunds, Dimitri Papageorgiou, and Rogelio Alvarez. Trilevel optimization in power network defense. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(4):712–718, 2007.

A Cut–Cocut Algebra

This section establishes basic algebraic properties of the set system operators introduced in Section 2. These operators arise frequently in the analysis of combinatorial structures (e.g., networks) and connect naturally to classical notions such as clutters and blockers [18, 21]. In particular, every clutter Ω is a set system consisting of minimal elements, and its blocker corresponds to $m(\mathcal{C}(\Omega))$ in our framework. The cut–cocut algebra developed here extends these ideas by incorporating the algebraic interactions between a larger family of operators (the eight operators in Definition 2).

While many of the results are intuitive, this algebra serves as a compact toolkit that streamlines later derivations in Appendix B and highlights the dual relationships among operators. We begin by formalizing the notion of duality for set system operators.

Definition 7 (Dual Operator)**.** Given a set system operator $g : \mathscr{P}(\mathscr{P}(\Delta)) \to \mathscr{P}(\mathscr{P}(\Delta))$, its *dual operator* g' is defined by

$$g'(\Omega) \;:=\; \{\, T \subseteq \Delta \;\mid\; \Delta \setminus T \in g(\widehat{\Omega}) \,\} \;=\; \widehat{g(\widehat{\Omega})}\,,$$

where $\widehat{(\cdot)}$ denotes the element–complement operator on set systems, i.e., $\widehat{\Omega} := \{\, \Delta \setminus T : T \in \Omega \,\}$.

Intuition. The dual g' applies g on the elementwise complement space $\widehat{\Omega}$ and then flips elements back by complementing again. Equivalently, $T \in g'(\Omega)$ if and only if the complement of T belongs to $g(\widehat{\Omega})$. Because elementwise complement is an involution, i.e., $\widehat{\widehat{\Omega}} = \Omega$, this duality on operators is symmetric.

Lemma 2. *If g' is the dual of g, then $(g')' = g$.*

Proof. Using the identity $g'(\cdot) = \widehat{g(\widehat{\cdot})}$, we obtain

$$(g')'(\Omega) \;=\; \widehat{g'(\widehat{\Omega})} \;=\; \widehat{\widehat{g(\widehat{\widehat{\Omega}})}} \;=\; g(\Omega),$$

where we used $\widehat{\widehat{\Omega}} = \Omega$ and the fact that $\widehat{(\cdot)}$ is an involution on set systems. □

A direct verification shows that $(\uparrow, \downarrow)$, (m, M), and $(\mathcal{C}, \mathcal{G})$ from Definition 2 are dual pairs. The next lemma captures the basic anti-commutativity between dual operators and elementwise complement; it will be used repeatedly.

Lemma 3. *For any dual pair g, g' and any set system Ω,*

$$\widehat{g(\Omega)} \;=\; g'(\widehat{\Omega}) \qquad \textit{and} \qquad \widehat{g'(\Omega)} \;=\; g(\widehat{\Omega}).$$

Proof. By definition, $g'(\widehat{\Omega}) = \widehat{g(\widehat{\widehat{\Omega}})} = \widehat{g(\Omega)}$. The second identity can be obtained by swapping g and g' and applying Lemma 2. □

Operator	Basic Properties	Dual Operator	Interactions
Complement $\overline{(\cdot)}$	Order-reversing; $\overline{\overline{\Omega}} = \Omega$; swaps upper/lower systems	Self-dual	$\overline{\widehat{\Omega}} = \widehat{\overline{\Omega}}$
Element-complement $\widehat{(\cdot)}$	Order-preserving; $\widehat{\widehat{\Omega}} = \Omega$; swaps upper/lower systems	Self-dual	Anticommutes with dual operators
Cut $\mathcal{C}(\cdot)$	Always upper; $\mathcal{C}(\mathcal{C}(\Omega)) =\uparrow \Omega$; depends only on minimal sets	Cocut $\mathcal{G}(\cdot)$	$\mathcal{C}(\widehat{\Omega}) = \widehat{\mathcal{G}(\Omega)}$
Cocut $\mathcal{G}(\cdot)$	Always lower; $\mathcal{G}(\mathcal{G}(\Omega)) =\downarrow \Omega$; depends only on maximal sets	Cut $\mathcal{C}(\cdot)$	$\mathcal{G}(\widehat{\Omega}) = \widehat{\mathcal{C}(\Omega)}$

Table 5: Summary of set system operators and their algebraic properties.

Equipped with these preliminaries, we collect the basic properties for the complement, element–complement, cut, and cocut operators, along with their interactions in the following theorem. A summary is provided in Table 5.

Theorem 5. *For any $\Omega \subseteq \mathscr{P}(\Delta)$, the following hold.*

***A*. *Complement* $\overline{(\cdot)}$.**

$\mathscr{A}$1. $\overline{\varnothing} = \mathscr{P}(\Delta)$ *and* $\overline{\mathscr{P}(\Delta)} = \varnothing$.

$\mathscr{A}$2. $\Omega \subseteq \Omega'$ *iff* $\overline{\Omega} \supseteq \overline{\Omega'}$ *(order-reversing).*

$\mathscr{A}$3. *If* Ω *is upper (resp. lower), then* $\overline{\Omega}$ *is lower (resp. upper).*

$\mathscr{A}$4. $\overline{\overline{\Omega}} = \Omega$ *(involution).*

***B*. *Element–complement* $\widehat{(\cdot)}$.**

$\mathscr{B}$1. $\widehat{\varnothing} = \varnothing$ *and* $\widehat{\mathscr{P}(\Delta)} = \mathscr{P}(\Delta)$.

$\mathscr{B}$2. $\Omega \subseteq \Omega'$ *iff* $\widehat{\Omega} \subseteq \widehat{\Omega}'$ *(order-preserving).*

$\mathscr{B}$3. *If* Ω *is upper (resp. lower), then* $\widehat{\Omega}$ *is lower (resp. upper).*

$\mathscr{B}$4. $\widehat{\widehat{\Omega}} = \Omega$ *(involution).*

***C*. *Cut* $\mathcal{C}(\cdot)$.**

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

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

$\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).*

D. Cocut $\mathcal{G}(\cdot)$***.***

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

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

$\mathscr{D}3$. *$\mathcal{G}(\Omega)$ 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).*

E. Interactions.

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

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

$\mathscr{E}3$. *$\widehat{M(\Omega)} = m(\widehat{\Omega})$ and $\widehat{m(\Omega)} = M(\widehat{\Omega})$ (anti-commute).*

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

$\mathscr{E}5$. *$\overline{\uparrow \Omega} \subseteq \downarrow \overline{\Omega}$ and $\overline{\downarrow \Omega} \subseteq \uparrow \overline{\Omega}$ (partial anti-commute).*

$\mathscr{E}6$. *$m(\overline{\Omega}) \subseteq \overline{M(\Omega)}$ and $M(\overline{\Omega}) \subseteq \overline{m(\Omega)}$ (partial anti-commute).*

$\mathscr{E}7$. *$\mathcal{G}(\overline{\Omega}) \subseteq \overline{\mathcal{C}(\Omega)}$ and $\mathcal{C}(\overline{\Omega}) \subseteq \overline{\mathcal{G}(\Omega)}$, with equality iff Ω is upper and lower, respectively (partial anti-commute).*

Proof. **(A)** Items ($\mathscr{A}1$), ($\mathscr{A}2$), ($\mathscr{A}4$) are immediate. For ($\mathscr{A}3$), if Ω is upper and $T \in \overline{\Omega}$, then any $T' \subsetneq T$ cannot lie in Ω (else upperness would force $T \in \Omega$), hence $T' \in \overline{\Omega}$, showing $\overline{\Omega}$ is lower. The reverse direction is symmetric.

(B) Items ($\mathscr{B}1$), ($\mathscr{B}2$), ($\mathscr{B}4$) are immediate from the definition of $\widehat{(\cdot)}$ as an involutive bijection on elements; ($\mathscr{B}3$) follows since complement reverses inclusion on elements.

(C) Items ($\mathscr{C}1$)–($\mathscr{C}5$) are standard from the definition of cut operator. For ($\mathscr{C}6$), if $\varnothing \in \Omega$ then $\mathcal{C}(\Omega) = \varnothing$ and $\mathcal{C}(\mathcal{C}(\Omega)) = \mathscr{P}(\Delta) = \uparrow \Omega$. Otherwise, $U \in \mathcal{C}(\mathcal{C}(\Omega))$ iff U intersects every S that intersects every $T \in \Omega$, which is equivalent to $U \supseteq T$ for some $T \in \Omega$, i.e., $U \in \uparrow \Omega$.

(E) For ($\mathscr{E}1$), $\widehat{\overline{\Omega}} = \{\, \Delta \setminus T : T \notin \Omega \,\} = \overline{\{\, \Delta \setminus T : T \in \Omega \,\}} = \overline{\widehat{\Omega}}$. Identities ($\mathscr{E}2$)–($\mathscr{E}4$) follow from Lemma 3 using that $(\uparrow, \downarrow)$ and $(\mathcal{C}, \mathcal{G})$ are dual pairs. For ($\mathscr{E}5$), $T \in \overline{\uparrow \Omega} \Rightarrow T \notin \uparrow \Omega \Rightarrow T \notin \Omega \Rightarrow T \in \overline{\Omega} \Rightarrow T \in \downarrow \overline{\Omega}$. The second inclusion is analogous. For ($\mathscr{E}6$), $T \in m(\overline{\Omega})$ implies $T \notin M(\Omega)$, hence $T \in \overline{M(\Omega)}$ (and symmetrically for the other inclusion). For ($\mathscr{E}7$), if $T \in \mathcal{G}(\overline{\Omega})$, then for all $T' \notin \Omega$ we have $T \cup T' \neq \Delta$, which implies $\Delta \setminus T \in \Omega$, hence $T \notin \mathcal{C}(\Omega)$ and $\mathcal{G}(\overline{\Omega}) \subseteq \overline{\mathcal{C}(\Omega)}$. 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

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

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

which completes the proof of necessity.

(D) Follows from the identities in (E) and the properties in (C) by duality. For instance, $(\mathscr{D}1)$–$(\mathscr{D}2)$ follow from $(\mathscr{C}1)$–$(\mathscr{C}2)$ via $\mathcal{G}(\cdot) = \widehat{\mathcal{C}(\widehat{\cdot})}$; $(\mathscr{D}3)$ from $(\mathscr{C}3)$; $(\mathscr{D}4)$ from $(\mathscr{C}4)$ together with $\widehat{m(\widehat{\Omega})} = M(\Omega)$; $(\mathscr{D}5)$ from $(\mathscr{C}5)$ using that $\widehat{(\cdot)}$ is order-preserving on set systems; and $(\mathscr{D}6)$ from $(\mathscr{C}6)$ plus $\uparrow \widehat{\Omega} = \widehat{\downarrow \Omega}$. □

B Mathematical Proofs

Theorem 1 (Upper Approximation)**.** *For every set system* Ω*, we have*

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

where $\mathcal{C}(\widehat{\overline{\Omega}})$ *and* $\mathcal{C}(\widehat{\overline{\uparrow \Omega}})$ *are the tightest inner and outer approximations of* Ω *using upper-systems, respectively. Moreover, equality holds throughout if and only if* Ω *is upper-closed.*

Proof. For the first inclusion, take any $T \in \mathcal{C}(\widehat{\overline{\Omega}})$ 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, $T \in \mathcal{C}(\widehat{\overline{\Omega}})$ means $T \cap S \neq \varnothing$ for all $S \in \widehat{\overline{\Omega}}$. In particular, $T \cap (\Delta \setminus T) \neq \varnothing$, a contradiction.

To prove $\mathcal{C}(\widehat{\overline{\Omega}})$ is the largest inner approximation, we show that any upper system $\Omega' \subseteq \Omega$ is a subset of $\mathcal{C}(\widehat{\overline{\Omega}})$. Suppose otherwise, take $T \in \Omega' \subseteq \Omega$ but $T \notin \mathcal{C}(\widehat{\overline{\Omega}})$. The latter implies that for some $S \in \widehat{\overline{\Omega}}$, we have $T \subseteq \Delta \setminus S \in \overline{\Omega}$, where the membership also implies $\Delta \setminus S \notin \Omega$. However, since Ω' is an upper system, we have $\Delta \setminus S \in \Omega' \subseteq \Omega$, a contradiction.

For the second inclusion, $\uparrow \Omega$ is clearly the tightest outer approximation of Ω by definition. Applying the above argument with the system $\uparrow \Omega$, we obtain

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

where the equality is due to $\uparrow \Omega$ is upper-closed.

Finally, since the systems on both sides are upper-closed, the equality holds on both sides if and only if Ω is also upper-closed. □

Corollary 2. *Given a set system* Ω*, the tightest inner approximation by upper-systems 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 inner approximations by lower-systems with equality conditions reversed.

Proof. Replacing Ω in the first inclusion of Theorem 1 with each of the right-side sets, we can derive these claims using the commutativity rule ($\mathscr{E}1$) and the self-inverse rules ($\mathscr{A}4$) and ($\mathscr{B}4$). Moreover, they are the tightest upper embeddings by Theorem 1. □

Lemma 1. *Given any set system Ω, the vector representations of $\mathcal{C}(\Omega)$ and $\widehat{\mathcal{C}(\Omega)}$ correspond to the following* covering *and* elimination *inequalities, respectively.*

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

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 $S \in \widehat{\mathcal{C}(\Omega)}$ 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 2. *Given a general BIP $\min_{x \in \mathcal{X}} f(x)$, let $\Omega := \{T_x \mid x \in \mathcal{X}\}$. The following reformulations serve as inner/outer approximations of the original problem,*

Upper-Inner Approximation Π_{ui}:

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

Upper-Outer Approximation Π_{uo}:

$$\min_{x \in \{0,1\}^n} \; f(x) \tag{2a}$$
$$s.t. \; \sum_{i \in T} x_i \geq 1, \; \forall T \in m\left(\widehat{\uparrow \Omega}\right) \tag{2b}$$

Lower-Inner Approximation Π_{li}:

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

Lower-Outer Approximation Π_{lo}:

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

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, (1) and (2) are both equivalent to the original problem if and only if Ω is upper-closed; (3) and (4) are both equivalent to the original problem if and only if Ω is a lower-closed.

Proof. These approximation results directly follow Theorem 1, Corollary 1, and Lemma 1. The value relationships follow the definition of inner and outer approximations. □

Corollary 3. *Suppose Ω is monotone with a membership oracle of complexity $O(\tau(\Omega))$. For any infeasible binary solution $x \in \{0,1\}^n$, the constraint separation complexity for (1)–(4) is of order $O(\log n \cdot \tau(\Omega))$.*

Proof. Suppose x is feasible, we are done. Otherwise, the separation algorithm needs to obtain an extremal (e.g., minimal if Ω is upper and maximal if Ω is lower) infeasible solution to prevent the current infeasible structure T_x, which can be done by a binary search with at most n elements. This proves the claimed complexity. □

Corollary 4. *Given a monotone system Ω, the following relationships hold,*

$$\text{When } \Omega \text{ is upper-closed: } \mathcal{P}_\Omega \subseteq \mathcal{P}^{\mathcal{C}}_{\widehat{\widehat{\Omega}}}, \quad \mathcal{P}_{\widehat{\widehat{\Omega}}} \subseteq \mathcal{P}^{\mathcal{C}}_{\Omega}$$

$$\text{When } \Omega \text{ is lower-closed: } \mathcal{P}_\Omega \subseteq \mathcal{P}^{\mathcal{G}}_{\overline{\overline{\Omega}}}, \quad \mathcal{P}_{\overline{\overline{\Omega}}} \subseteq \mathcal{P}^{\mathcal{G}}_{\Omega}.$$

Moreover, we have $\min_{x\in\mathcal{P}_\Omega, y\in\mathcal{P}_{\widehat{\widehat{\Omega}}}} \langle x, y\rangle \geq 1$ *and* $\max_{x\in\mathcal{P}_\Omega, y\in\mathcal{P}_{\overline{\overline{\Omega}}}} \langle 1, x+y\rangle - \langle x, y\rangle \leq n-1$ *for the two cases, respectively.*

Proof. The first inclusion for the upper system Ω follows directly from Theorem 2: because Formulation (1) is an equivalent reformulation of $\min_{x\in\mathcal{X}_\Omega} f(x)$, its linear relaxation must contain the convex hull of $\mathcal{X}_\Omega$. The second inclusion is due to the identity $\mathcal{C}(\Omega) = \widehat{\widehat{\Omega}}$ from Corollary 2 when Ω is upper. Then, by the same reason, the relaxation of cutting all elements in Ω contains the convex hull of $\mathcal{X}_{\widehat{\widehat{\Omega}}}$. A symmetric argument proves the two inclusion relationships for a lower-closed Ω.

For the bound of the first inner product, it is true by assumption ($+\infty$ if the solution space is empty in minimization) when the solution space is empty, i.e., when $\Omega \in \{\varnothing, \mathscr{P}(\Delta)\}$. Otherwise, every $x \in \mathcal{P}_\Omega$ is the convex combination of some extreme points of $\mathcal{X}_\Omega$, which can be written as $\sum_{i\in I} \lambda_i x_i$. Similarly, we represent y as the convex combination $\sum_{j\in J} \gamma_j y_j$ of some extreme points $y_j \in \mathcal{P}_{\widehat{\widehat{\Omega}}}$, which leads to the following

$$\langle x, y\rangle = \sum_{(i,j)\in I\times J} \lambda_i \gamma_j \langle x_i, y_j\rangle.$$

Since $\Omega = \mathcal{C}(\widehat{\widehat{\Omega}})$ when Ω is upper-closed, every pair of x_i and y_j is binary and must have non-empty intersection by the definition of cut operator $\mathcal{C}$, implying $\langle x_i, y_j\rangle \geq 1$ for every $(i,j) \in I\times J$. Thus, $\langle x, y\rangle$ is a convex combination of scalars that are at least one, proving the desired inequality. □

Theorem 3. *Given a binary solution space $\mathcal{X} := \{x \in \{0,1\}^n \mid g(x) \leq 0\}$ and an index set $I \subseteq [n]$, define the flipping map $\theta_I : \{0,1\}^n \to \{0,1\}^n$ by*

$$(\theta_I(x))_i := \begin{cases} 1 - x_i & \textit{if } i \in I, \\ x_i & \textit{otherwise.} \end{cases}$$

Let $\mathcal{X}_{\theta_I} := \{x \in \{0,1\}^n \mid g(\theta_I(x)) \le 0\}$*. Then,* θ_I *induces a bijective affine map between* $\operatorname{conv}(\mathcal{X})$ *and* $\operatorname{conv}(\mathcal{X}_{\theta_I})$*. In particular, an inequality* $\langle a, x\rangle \ge b$ *is valid (or facet-defining) for* $\operatorname{conv}(\mathcal{X})$ *if and only if* $\langle a, \theta_I(x)\rangle \ge b$ *is valid (or facet-defining) for* $\operatorname{conv}(\mathcal{X}_{\theta_I})$*.*

Proof. For every fixed index subset $I \subseteq [n]$, the transformation $\theta_I(\cdot)$ is clearly affine and injective by definition. To show θ_I is also surjective, we represent an arbitrary $x' \in \operatorname{conv}(\mathcal{X}_{\theta_I})$ as the convex combination $\sum_{j\in J} \lambda_j x'_j$ for some extreme points x'_j of $\operatorname{conv}(\mathcal{X}_{\theta_I})$ and define $x = \sum_{j\in J} \lambda_j \theta_I(x'_j)$. Clearly, we have $\theta_I(x) = x'$ since θ_I is affine and is an involution. We left to show $x \in \mathcal{X}$. By construction, each x'_j is binary and satisfies $g(\theta_I(x'_j)) \le 0$, implying $\theta_I(x'_j) \in \mathcal{X}$. Hence, the convex combination x belongs to the convex hull $\operatorname{conv}(\mathcal{X})$. Since affine bijections preserve faces and their dimensions, the statement follows. □

Corollary 5. *Suppose* Ω *is an upper system and* $|T| \ge 2$ *for every* $T \in m(\widehat{\overline{\Omega}})$*, the covering inequality* (1b) *is facet-defining if and only if* $\Delta \setminus T \in M(\overline{\Omega})$ *is quasi-feasible.*

Proof. This statement is a paraphrase of the Statements 1 and 5 in [3] (from the matrix perspective) and Proposition 3.4 in [56] (from the set system perspective). According to the latter, given an upper system Ω such that $|T| \ge 2$ for every $T \in m(\widehat{\overline{\Omega}})$, (1b) is facet-defining in the associated set cover problem 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. □

Proposition 1. *The following functions are bimonotone:*

- *Linear (modular) functions.*
- *Bilinear functions* $\langle x, Ry\rangle$ *for some block-diagonal matrix* $R \in \mathbb{R}^{I\times J}$ *where there exist (possibly trivial) index bipartitions* $I = I_1 + I_2$ *and* $J = J_1 + J_2$ *such that the diagonal blocks satisfy* $R_{I_1J_1} \ge 0$ *and* $R_{I_2J_2} \le 0$ *(entrywise).*
- *Submodular functions* f *where for every* $i \in \Delta$*, either*

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

- *The supermodular counterpart.*

Proof. For every linear function $\langle c, x\rangle$ with index set $[n]$, defining $I := \{i \in [n] \mid c_i \ge 0\}$ induces the required index bipartition. The described bilinear functions with index set $I \cup J$ can be written as $\langle x_{I_1}, R_{I_1J_1} y_{J_1}\rangle + \langle x_{I_2}, R_{I_2J_2} y_{J_2}\rangle$. Since $R_{I_1J_1} \ge 0$ and $R_{I_2J_2} \le 0$ by assumption, the index bipartition $(I_1 \cup J_1, I_2 \cup J_2)$ satisfies the bimonotone requirement. For submodular functions, 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 $T \subseteq \Delta \setminus \{i\}$. Hence, defining $I := \{i \mid f(\Delta) - f(\Delta \setminus \{i\}) \geq 0\}$ induces the required index bipartition. The supermodular case has all above inequality reversed, thus the index bipartition induced by $I := \{i \mid f(\{i\}) - f(\varnothing) \geq 0\}$ proves the claim. $\square$

Theorem 4. *Given a BIP* $\min_{x=(x_I,x_J)\in\mathcal{X}_\Omega} f(x_I, x_J)$ *with a bimonotone objective* f *that is increasing and decreasing in* x_I *and* x_J*, respectively. Then, an equivalent reformulation is:*

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

$$\text{s.t.} \sum_{i\in T_I} x_i + \sum_{j\in J\setminus T_J} (1 - x_j) \geq 1, \ \forall (T_I, T_J) \in \mathcal{E}\left(\widehat{\uparrow_I\downarrow_J \Omega}\right). \tag{5b}$$

Moreover, suppose $|T_I \cup (J \setminus T_J)| \geq 2$ *for every* $(T_I, T_J) \in \widehat{\uparrow_I\downarrow_J \Omega}$*, the above inequality is facet-defining if and only if the following two conditions are both satisfied:*

- *For every* $i \in I \setminus T_I$*, there exists some* $i' \in T_I$ *such that* $((I \setminus T_I) \setminus \{i\} \cup \{i'\}, T_J) \in \uparrow_I\downarrow_J \Omega$*;*
- *For every* $j \in T_J$*, there exists some* $j' \in J \setminus T_J$ *such that* $(I \setminus T_I, T_J \setminus \{j\} \cup \{j'\}) \in \uparrow_I\downarrow_J \Omega$*.*

Proof. Using the flipping map θ_J (introduced in Theorem 3), we have $\mathcal{X}_\Omega = \theta_J(\theta_J(\mathcal{X}_\Omega))$. Then, the original optimization problem is equivalent to the following due to θ_J is an involution

$$\min_{x\in\theta_J(\mathcal{X}_\Omega)} f(\theta_J(x)).$$

Since f is increasing in x_I and decreasing in x_J, the new objective function $f \circ \theta_J$ is increasing over $\theta_J(\mathcal{X}_\Omega)$. Then, by Theorem 2, we can extend the solution space to be its upper-closure and obtains the following exact representation

$$\begin{aligned} \min_{x\in\{0,1\}^n} \ & f \circ \theta_J(x) \\ \text{s.t.} \ & \sum_{i\in T} x_i \geq 1, \quad \forall T \in m(\widehat{\uparrow \Omega_{\theta_J}}) \end{aligned}$$

where Ω_{θ_J} is defined as $\{T \mid x_T \in \theta_J(\mathcal{X}_\Omega)\}$. By the index bipartition (I, J), we can further express the above covering inequalities into

$$\sum_{i\in T_I} x_i + \sum_{j\in J\setminus T_J} x_j \geq 1, \quad \forall (T_I, J \setminus T_J) \in m(\widehat{\uparrow \Omega_{\theta_J}}).$$

We next show $(T_I, J \setminus T_J) \in m(\widehat{\uparrow \Omega_{\theta_J}})$ if and only if $(T_I, T_J) \in \mathcal{E}(\widehat{\uparrow_I\downarrow_J \Omega})$. By definition, we have

$$\Omega_{\theta_J} = \{T \mid \theta_J(x_T) \in \mathcal{X}_\Omega\} = \{(T_I, T_J) \mid (T_I, J \setminus T_J) \in \Omega\} = \{(T_I, J \setminus T_J) \mid (T_I, T_J) \in \Omega\},$$

which implies that the associated upper-closure $\uparrow \Omega_{\theta_J}$ can be computed as

$$\uparrow \Omega_{\theta_J} = \{(T_I, J \setminus T_J) \mid T_I \supseteq T_I' \text{ and } T_J \subseteq T_J' \text{ for some } (T_I', T_J') \in \Omega\}.$$

Thus, $(T_I, J \setminus T_J) \in \uparrow \Omega_{\theta_J}$ if and only if $(T_I, T_J) \in \uparrow_I \downarrow_J \Omega$ by the definition of bimonotone closure. Then, we obtain

$$(T_I, J \setminus T_J) \in \widehat{\overline{\uparrow \Omega_{\theta_J}}} \iff (I \setminus T_I, T_J) \in \overline{\uparrow \Omega_{\theta_J}} \iff (I \setminus T_I, J \setminus T_J) \in \overline{\uparrow_I \downarrow_J \Omega} \iff (T_I, T_J) \in \widehat{\overline{\uparrow_I \downarrow_J \Omega}}.$$

Moreover, the structure $(T_I, J \setminus T_J)$ is minimal in $\widehat{\overline{\uparrow \Omega_{\theta_J}}}$ if and only if its counterpart (T_I, T_J) is extremal in $\widehat{\overline{\uparrow_I \downarrow_J \Omega}}$. Thus, the covering inequalities can be equivalently expressed as

$$\sum_{i \in T_I} x_i + \sum_{j \in J \setminus T_J} x_j \geq 1, \quad \forall (T_I, T_J) \in \mathcal{E}(\widehat{\overline{\uparrow_I \downarrow_J \Omega}}). \tag{a}$$

Then, substituting x_j by $1 - x_j'$ for every $j \in J$ in objective function and all the constraints proves the correctness of (5).

According to Theorem 3, the facet condition for a bimonotone cut is equivalent to the one for the covering inequality (10). Then, the stated facet condition is simply the interpretation of Corollary 5 in this bimonotone setting for (10). $\square$

Corollary 6. *Given a constraint of a BIP defined by $g(x_I, x_J) \geq 0$ for some bimonotone function g that is increasing in x_I and decreasing in x_J, then all the structures satisfy this constraint form a set system Ω that is bimonotone, i.e., $\Omega = \uparrow_I \downarrow_J \Omega$. In particular, the associated constraint set (5b) provides a linear representation of $g(x_I, x_J) \geq 0$.*

Proof. For every structure $T = (T_I, T_J)$ satisfying the constraint $g(x_{T_I}, x_{T_J}) \geq 0$, adding additional elements from I to T_I or removing elements from T_J would increase the value of g since it is increasing in x_I and decreasing in x_J. This implies that the set of structures feasible to the constraint is bimonotone and satisfies $\Omega = \uparrow_I \downarrow_J \Omega$. By Theorem 4, the constraint set (5b) is an exact linear representation. $\square$

Proposition 2. *For any set system Ω, $\uparrow \Omega \cap \downarrow \Omega$ is the smallest interval system that contains Ω. Moreover, Ω is interval if and only if $\Omega = \uparrow \Omega \cap \downarrow \Omega$.*

Proof. It is immediate that $\Omega \subseteq \uparrow \Omega \cap \downarrow \Omega$. To show that the intersection forms an interval system, take $T, T' \in \uparrow \Omega \cap \downarrow \Omega$ and any T'' with $T \subseteq T'' \subseteq T'$. Since $T \in \uparrow \Omega$, there exists $S \in \Omega$ with $S \subseteq T \subseteq T''$, hence $T'' \in \uparrow \Omega$. Similarly, because $T' \in \downarrow \Omega$, there exists $S' \in \Omega$ with $T' \subseteq S'$, and thus $T'' \subseteq S'$, so $T'' \in \downarrow \Omega$. Therefore $T'' \in \uparrow \Omega \cap \downarrow \Omega$, proving closure under intervals.

For minimality, let Ω' be any interval system with $\Omega \subseteq \Omega'$. Take any $T \in \uparrow \Omega \cap \downarrow \Omega$. Then there exist $T_1, T_2 \in \Omega$ with $T_1 \subseteq T \subseteq T_2$. Since $\Omega \subseteq \Omega'$ and Ω' is interval, it follows that $T \in \Omega'$. Thus $\Omega' \supseteq \uparrow \Omega \cap \downarrow \Omega$, showing minimality.

For the second statement, if Ω is interval, then by minimality we must have $\Omega \supseteq \uparrow \Omega \cap \downarrow \Omega$, and the reverse inclusion holds trivially. Conversely, if $\Omega = \uparrow \Omega \cap \downarrow \Omega$, then Ω equals an interval system, hence itself is interval. $\square$

Proposition 3. *Every set system* Ω *adopts a decomposition* $\Omega = \bigcup_{k\in K} \Omega_k$ *for some interval set systems* $\{\Omega_k\}_{k\in K}$*, which leads to the following exact reformulation of the associated problem* $\min_{x\in\mathcal{X}_\Omega} f(x)$

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

$$s.t. \; \sum_{i\in T} x_i \geq z_k, \qquad \forall k \in K, \forall T \in m\left(\widehat{\overline{\uparrow \Omega_k}}\right), \tag{6b}$$

$$\sum_{i\in T} x_i \leq |T| - z_k, \; \forall k \in K, \forall T \in m\left(\overline{\downarrow \Omega_k}\right), \tag{6c}$$

$$\sum_{k\in K} z_k = 1, \tag{6d}$$

where z_k *represents the set system to which the optimal solution belongs.*

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. Then, each identified interval system can be represented by covering and elimination inequalities in (6b) and (6c), and binary variables z_k are used to choose exactly one interval system. □