Binary Integer Program Reformulation: A Set System Approximation Approach

Ningji Wei

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

Abstract

This paper presents a generic reformulation framework for binary integer programs (BIPs) without imposing additional specifications for the objective function or constraints. To facilitate such generality, we introduce a set system approximation theory designed to identify the tightest inner and outer approximations for any binary solution space using special types of set systems. This development leads to an exact reformulation framework for general BIPs, centered around set covering and subtour elimination inequalities. We investigate the implications of this methodology on various instance problems, uncovering new solution strategies and structural insights. Building upon these advancements, we also extend the classic max-flow min-cut theorem into a broader context of set system duality. Overall, our framework explores a new direction in the field of integer programming by examining the algebraic properties of set systems and their operators, which may spur additional research questions and further enrich the field.

Keywords: integer programming, reformulation, duality, set covering, subtour elimination

1 Introduction

Influential reformulation techniques have emerged in the history of *mixed integer programming* (MIP). These include Dantzig-Wolfe reformulation [8], Benders decomposition [4], polyhedron projection [14], dynamic programming reformulation [27], among others. Each of these methods offers a unique analytical perspective, exploring fundamental problem structures to develop distinct reformulation techniques. Over the past decades, these methods have been permeating into various application domains, yielding substantial theoretical and computational impacts.

This paper explores another generic reformulation method for the class of *binary integer program* (BIP) described as follows,

$$\Pi : \quad \min_{x \in \mathcal{X} \subseteq \{0,1\}^n} \; f(x), \tag{1}$$

where $\mathcal{X}$ is a general binary solution space and f is a general function over $\mathcal{X}$. Except the binary requirement on $\mathcal{X}$, we do not impose extra restrictions on the representation of f or $\mathcal{X}$. Due to such

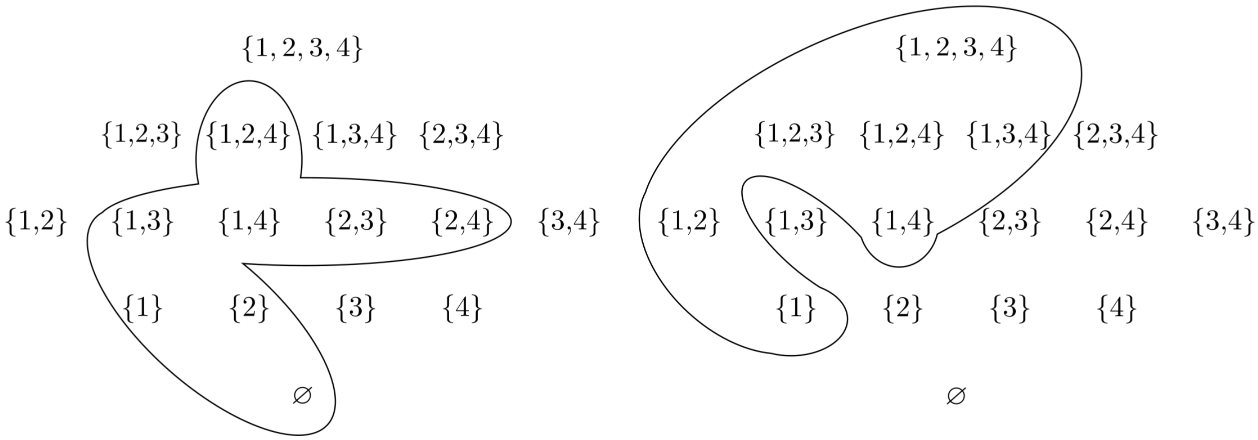

Figure 1: Two set systems, Ω_1 (left) and Ω_2 (right), each defined over the ground set $\Delta = \{1, 2, 3, 4\}$. Both set systems encompass all elements enclosed within their respective curves.

relaxed conditions, Formulation (1) can be used to describe a variety of interesting optimization problems, including but not limited to the following.

- Interdiction games with $\mathcal{Y}_x$ as the follower's decision space

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

- Multistage stochastic BIPs with ξ_i's as uncertain parameters

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

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

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

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

While our analytical approach markedly differs from traditional methodologies in the literature, the reformulations we derive demonstrate a close connection with two well-established types of constraints: the renowned set covering inequalities [2] and the widely recognized subtour elimination inequalities originated from the traveling salesman problem (TSP) [3].

1.1 Literature Review

Several generic reformulation techniques have been developed to transform MIPs into various forms, offering new insights and solution approaches. The Dantzig-Wolfe reformulation [8] focuses on using the extreme points and rays of the solution convex hull for rewriting feasible solutions via Minkowski representation [23, Theorem 13.1]. Benders decomposition [4] views the solution set as a continuous space parameterized by integer variables, leading to a suite of reformulation techniques based on various duality theories. The polyhedron projection method [14], employing Farkas' Lemma [10], aims to tighten solution space characterization by reducing auxiliary variables. Conversely, extended formulations [7] seek a more compact solution space description through the introduction of new variables. For problems with optimal substructures, dynamic programming reformulation [27] can be used to capture such nice properties effectively. For interested readers, we refer to [23] for a comprehensive coverage of this topic.

In addition to these well-established methodologies, two special types of inequalities have been notably prevalent across various instance problems with distinct characteristics. Set covering inequalities are widely employed in different contexts, such as interdiction games [13, 17, 26], vehicle routing [19], network design [12], power grid optimization [28], and facilitate location problems [5], often enhancing the branch-and-cut implementation framework. Moreover, these covering inequalities have shown to possess strong facet properties in various problem settings under certain conditions [20, 21, 24]. Similarly, subtour elimination inequalities have become a critical reformulation component in transportation [1, 11, 15], production scheduling [6], location-routing problems [16], and have demonstrated significant impact in computational efficiency when properly strengthened and implemented [9, 22].

The widespread discovery and application of these two types of inequalities in diverse instance problems suggest the necessity for a new theory, which would unify these inequalities within a universal framework. As we will demonstrate later, such a framework emerges naturally from the proposed set system approximation theory.

1.2 Contributions

A primary contribution of this project is the following exact reformulation of the general BIP (1). Its detailed derivation and analysis can be found in Section 4.

$$\min_{x\in\{0,1\}^n, z\in\{0,1\}^{|K|}} \ f(x) \tag{2a}$$

$$\text{s.t.} \sum_{i\in T} x_i \geq z_k, \qquad \forall k\in K, \forall T\in \Omega_k^u, \tag{2b}$$

$$\sum_{i\in S} x_i \leq |S| - z_k, \ \forall k\in K, \forall S\in \Omega_k^l, \tag{2c}$$

$$\sum_{k\in K} z_k = 1. \tag{2d}$$

This reformulation, which utilizes the upper and lower set systems in $\{\Omega^u_k, \Omega^l_k\}_{k\in K}$ to characterize the solution space $\mathcal{X}$, leads to several interesting implications. First, regardless of their original description for $\mathcal{X}$, every BIP can be reformulated using only the covering inequalities (2b), elimination inequalities (2c), along with a choice constraint (2d), suitable for a branch-and-cut implementation. Second, this formulation supports a parallel implementation, where each computational session solves one optimization problem with a distinct vector z. Moreover, the best incumbent solution across all sessions can be shared to improve each individual computation. Third, existing facet analysis and constraint strengthening methods regarding covering and elimination constraints can be immediately transported to study various BIPs, which also brings abundant valid and supervalid constraints that can potentially enhance existing solution methods. Finally, this unified approach enables the comparative analysis across different BIPs by investigating their associated structures in $\{\Omega^u_k, \Omega^l_k\}_{k\in K}$. In addition to this key result, our contributions also include the following.

- We develop a set system approximation theory that identifies the tightest inner and outer approximations for any given set system using upper and lower systems. To enable this analysis, we also establish a set of algebraic rules between several set system operators.

- We convert the set system approximation results into four inner and outer approximation reformulations for general BIPs and identify the conditions for them to be exact. We further extend the application scope of these reformulation techniques by exploring the objective function's properties and developing a decomposition scheme for arbitrary set systems.

- We illustrate our framework in multiple BIP instances, which leads to new formulations for solving various optimization problems, such as the longest simple path problem and BIPs with distributionally robust chance constraints.

- We also develop a duality theory for set systems that generalize the classic flow-cut duality relationship. Based on this sharp result, we identify dual structural pairs in various graph structures and set systems.

The presentation of our results is organized as follows. In Section 2, we establish the fundamental set system approximation theory. In Section 3, we delve into reformulation approximation techniques and assess their separation subroutines and runtime complexity. Section 4 introduces methods to extend the reformulation framework for arbitrary set systems. In Section 5, we generalize the classic flow-cut duality to arbitrary set systems and identify such structural dual pairs in various problem settings. Finally, Section 6 concludes with key remarks and a summary of our findings.

2 Set System Approximation

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

2.1 Preliminaries

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

2.1.1 Set System

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

2.1.2 Structure Operator & Set System Operator

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

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

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

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

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

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

$f \circ g = g \circ f$, and call them anticommutative if

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

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

2.1.3 Relevant Set Systems and Operators

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

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

- $\uparrow \Omega := \{T \mid T \supseteq T' \text{ for some } T' \in \Omega\}$,
- $\downarrow \Omega := \{T \mid T \subseteq T' \text{ for some } T' \in \Omega\}$,
- $m(\Omega) := \{T \in \Omega \mid \forall T' \in \Omega, T' \subseteq T \implies T' = T\}$,
- $M(\Omega) := \{T \in \Omega \mid \forall T' \in \Omega, T' \supseteq T \implies T' = T\}$.

These four operators are termed the up-closure, down-closure, minimal operator, and maximal operator, respectively.

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

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

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

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

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

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

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

These operators are termed the complement operator, element-complement operator, cut operator, cocut operator, and elimination operator, respectively.

These operators have intuitive interpretations. Operator $\overline{(\cdot)}$ is designed to return all the non-solutions or non-structures within $\mathscr{P}(\Delta)$. Operator $\widehat{(\cdot)}$ is commonly applied in graph theory to generate the complement subgraph. For instance, suppose Ω is the set of independent sets in a network, then $\widehat{\Omega}$ contains all the vertex covers. We name $\mathcal{C}$ the cut operator because every element in $\mathcal{C}(\Omega)$ intersects every structure $T \in \Omega$, which mimics the relationship between s-t edge cuts and s-t paths in a given network. The significance of this operator in interdiction games has been studied in [25]. Clearly, the cut operator $\mathcal{C}$ and cocut operator $\mathcal{G}$ form a dual pair. The cocut operator can also be equivalently defined as follows,

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

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

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

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

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

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

2.2 Cut-Cocut Algebra

We derive some basic algebraic properties for the previously defined operations and organize them into five parts. The first four explore the basic properties associated with the first four operators in Definition 2, and the last part lists the corresponding interactions.

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

- $\mathscr{A}$. *For the complement operator $\overline{(\cdot)}$,*
  - $\mathscr{A}$*1.* $\overline{\varnothing} = \mathscr{P}(\Delta)$ *and* $\overline{\mathscr{P}(\Delta)} = \varnothing$
  - $\mathscr{A}$*2.* Ω *and* $\overline{\Omega}$ *form a partition of* $\mathscr{P}(\Delta)$
  - $\mathscr{A}$*3.* $\Omega \subseteq \Omega'$ *if and only if* $\overline{\Omega} \supseteq \overline{\Omega'}$ *(order-reversing)*
  - $\mathscr{A}$*4.* Ω *is an upper (lower) system implies* $\overline{\Omega}$ *is a lower (upper) system*
  - $\mathscr{A}$*5.* $\overline{\overline{\Omega}} = \Omega$ *(self-inverse)*
- $\mathscr{B}$. *For the element-complement operator $\widehat{(\cdot)}$,*
  - $\mathscr{B}$*1.* $\widehat{\varnothing} = \varnothing$ *and* $\widehat{\mathscr{P}(\Delta)} = \mathscr{P}(\Delta)$
  - $\mathscr{B}$*2.* $\Omega \subseteq \Omega'$ *if and only if* $\widehat{\Omega} \subseteq \widehat{\Omega'}$ *(order-preserving)*
  - $\mathscr{B}$*3.* $\widehat{(\cdot)}$ *acts as a contravariant functor on Ω, i.e., for every $T, T' \in \Omega$ such that $T \subseteq T'$, we have $\Delta \setminus T, \Delta \setminus T' \in \widehat{\Omega}$ with inclusion direction reversed*
  - $\mathscr{B}$*4.* Ω *is an upper (lower) system implies* $\widehat{\Omega}$ *is a lower (upper) system*
  - $\mathscr{B}$*5.* $\widehat{\widehat{\Omega}} = \Omega$ *(self-inverse)*
- $\mathscr{C}$. *For the cut operator $\mathcal{C}(\cdot)$,*
  - $\mathscr{C}$*1.* $\mathcal{C}(\varnothing) = \mathscr{P}(\Delta)$ *and* $\mathcal{C}(\mathscr{P}(\Delta)) = \varnothing$
  - $\mathscr{C}$*2.* $\mathcal{C}(\Omega) = \varnothing$ *if and only if* $\varnothing \in \Omega$
  - $\mathscr{C}$*3.* $\mathcal{C}(\Omega)$ *is an upper system*
  - $\mathscr{C}$*4.* $\mathcal{C}(\Omega) = \mathcal{C}(m(\Omega))$
  - $\mathscr{C}$*5.* $\Omega \subseteq \Omega'$ *implies* $\mathcal{C}(\Omega) \supseteq \mathcal{C}(\Omega')$ *(order-reversing)*
  - $\mathscr{C}$*6.* $\mathcal{C}(\mathcal{C}(\Omega)) = \uparrow \Omega$ *(upper envelope)*
- $\mathscr{D}$. *For the cocut operator $\mathcal{G}(\cdot)$,*
  - $\mathscr{D}$*1.* $\mathcal{G}(\varnothing) = \mathscr{P}(\Delta)$ *and* $\mathcal{G}(\mathscr{P}(\Delta)) = \varnothing$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

where each equivalence is from the definitions. The proofs for ($\mathscr{E}$2)–($\mathscr{E}$4) can be unified utilizing the property of the contravariant functor $\widehat{(\cdot)}$, which we postpone to show in Proposition 2. For

($\mathscr{E}$5), we have

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

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

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

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

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

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

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

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

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

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

$$\mathcal{C}(\widehat{\Omega}) = \mathcal{C}(m(\widehat{\Omega})) \Longrightarrow \mathcal{G}(\Omega) = \widehat{\mathcal{C}(\widehat{\Omega})} = \widehat{\mathcal{C}(m(\widehat{\Omega}))} = \mathcal{G}\left(\widehat{m(\widehat{\Omega})}\right) = \mathcal{G}\left(M\left(\widehat{\widehat{\Omega}}\right)\right) = \mathcal{G}(M(\Omega)).$$

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

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

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

To complete the proof of Theorem 1, we need the following proposition to show the anticommutativity of $\widehat{(\cdot)}$ with respect to dual operator pairs.

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

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

Proof. By ($\mathscr{B}$3), $\widehat{(\cdot)}$ is a contravariant functor on $\mathscr{P}(\Delta)$, i.e., $\widehat{\mathscr{P}(\Delta)}$ can be considered the same as $\mathscr{P}(\Delta)$ with all the inclusion direction reversed. Moreover, from the definition of dual pair, all the substitution rules will replace one thing in $\mathscr{P}(\Delta)$ with its contravariant counterpart in $\widehat{\mathscr{P}(\Delta)}$. Thus, for every dual operator pair g, g', we have

$$S \in g(\Omega) \Longleftrightarrow \Delta \setminus S \in g'(\Omega) \Longleftrightarrow S \in \widehat{g'(\Omega)},$$

which concludes the proof. □

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

Corollary 1. *We have the following identities:*

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

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

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

2.3 Set System Approximation

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

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

Lemma 1 (Upper Embedding)**.** *For any set system $\Omega \subseteq \mathscr{P}(\Delta)$, we have*

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

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

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

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

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

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

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

Proof. The first two equalities are obvious since we have

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

by ($\mathscr{E}3$), and the equality follows ($\mathscr{C}4$) and ($\mathscr{C}5$). The relation $\Omega \subseteq \mathcal{C}^2(\Omega) = \uparrow \Omega$ is a direct consequence of ($\mathscr{C}6$). The other inclusion is implied by Lemma 1 and ($\mathscr{C}4$). To prove $\mathcal{C}\left(m\left(\widehat{\overline{\Omega}}\right)\right)$

is the tightest embedding, consider any upper set $\Omega' \subseteq \Omega$, we show that

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

Suppose otherwise, take $T \in \Omega' \subseteq \Omega$ but $T \notin \mathcal{C}\left(\widehat{M(\overline{\Omega})}\right)$. The latter implies that for some $S \in \widehat{M(\overline{\Omega})}$, we have $T \subseteq \Delta \setminus S \in M(\overline{\Omega})$, where the membership also implies $\Delta \setminus S \notin \Omega$. However, since Ω' is an upper set, we have $\Delta \setminus S \in \Omega' \subseteq \Omega$, leading to the desired contradiction. For $\uparrow \Omega$, it is obvious from the definition that it is the tightest upper enclosing of Ω. Furthermore, the two inclusions become equalities when Ω is an upper system by Lemma 1 and ($\mathscr{C}$6). Finally, for the last equality, we use the relation

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

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

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

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

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

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

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

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

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

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

Proof. Replacing Ω in the main result of Lemma 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}$5) and ($\mathscr{B}$5). They are the tightest upper embeddings due to Theorem 2 □

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

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

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

as the upper/lower approximation operators.

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

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

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

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

while the tightest lower embedding and enclosing are the trivial set systems. $\triangle$

3 Approximation Reformulations for BIPs

In this section, we will establish a reformulation framework for the general BIP (1) utilizing the set system approximation results. We will also develop general constraint separation subroutines for these reformulations and analyze their time complexity. Throughout this section, we consider $\Omega := \Omega_{\mathcal{X}}$ as the set system corresponding to the feasible region $\mathcal{X}$ of Problem (1).

3.1 Approximation Reformulations

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

Lemma 2. *Given any set system Ω, the vector representations of $\mathcal{C}(\Omega)$ and $\mathcal{E}(\Omega) = \mathcal{G}(\widehat{\Omega})$ are equivalent to the following* covering *and* elimination *inequalities, respectively.*

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

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

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

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

Theorem 3. *Any general BIP (1) can be reformulated into the following forms for inner/outer approximations,*

Upper Embedding Π_{ui} :

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

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

Upper Enclosing Π_{uo} :

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

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

Lower Embedding Π_{li} :

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

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

Lower Enclosing Π_{lo} :

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

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

with objective values satisfying

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

Moreover, (3) and (4) are both equivalent to (1) if and only if Ω is an upper system; (5) and (6) are both equivalent to (1) if and only if Ω is a lower system.

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

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

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

- Π_{ui} *is infeasible if and only if $\Delta \notin \Omega$, and is redundant if and only if $\Omega = \mathscr{P}(\Delta)$;*

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

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

In practical applications, the covering and elimination constraint sets in Lemma 2 usually consist of a considerable number of inequalities. This requires the use of an iterative cut generation approach for their separation. In the following subsection, we will introduce general separation subroutines and conduct an analysis of the associated time complexity.

3.2 A General Separation Procedure

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

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

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

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

Theorem 4. *Given an upper system* $\Omega \neq \varnothing$ *as the solution space of* (3)*, a structure* $T_x \subseteq \Delta$*, and a chain* C *from* T_x *to* Δ*. Algorithm 1 returns* $\varnothing$ *whenever* T_x *is feasible, and returns a structure in* $\widehat{\overline{\Omega}}$ *associated with the maximum element in* $C \cap \overline{\Omega}$ *if otherwise. Let* $O(\tau(\Omega))$ *be the*

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

input : the ground set Δ, a membership oracle $\mathbb{I}_\Omega$, current solution T_x
output: a structure $T \in \widehat{\widehat{\Omega}}$ or $\varnothing$

```
if I_Ω(T_x) = 1 then return ∅
C ← any inclusion chain from T_x to Δ
S ← BinarySearch(C, I_Ω)
return Δ \ S
```

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

input : an inclusion chain C, a membership oracle $\mathbb{I}_\Omega$
output: the maximum structure $S \in C \cap \overline{\Omega}$

```
if |C| = 1 then return the only element in the chain C
k ← Int(|C|/2)
if I_Ω(C[k]) = 1 then
│ C ← C[: k − 1]
else
│ C ← C[k :]
return BinarySearch(C, I_Ω)
```

complexity of the membership oracle $\mathbb{I}_\Omega$*, the complexity for separating such an inequality in* (3b) *is* $O(\log(|\Delta| - |T_x| + 1) \cdot \tau(\Omega))$.

Proof. Constraint (3b) is equivalent to

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

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

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

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

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

input : the ground set Δ, a membership oracle $\mathbb{I}_\Omega$, current solution T_x
output: a structure $T \in \overline{\Omega}$ or $\varnothing$

if $\mathbb{I}_\Omega(T_x) = 1$ **then return** $\varnothing$
$C \leftarrow$ any inclusion chain from $\varnothing$ to T_x
return `BinarySearch`$(C, \mathbb{I}_\Omega)$

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

input : an inclusion chain C, a membership oracle $\mathbb{I}_\Omega$
output: the minimum structure $S \in C \cap \overline{\Omega}$

if $|C| = 1$ **then return** the only element in the chain C
$k \leftarrow$ `Int`$(|C|/2)$
if $\mathbb{I}_\Omega(C[k]) = 1$ **then**
  $C \leftarrow C[k+1:]$
else
  $C \leftarrow C[:k]$
return `BinarySearch`$(C, \mathbb{I}_\Omega)$

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

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

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

3.3 Examples

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

Since all examples in this subsection are related to graphs, we will introduce the basic notation set. Given a graph G, $V(G)$ and $E(G)$ are the corresponding vertex and edge sets, which will be denoted simply by V and E when the underlying graph G is clear from the context. Given any vertex and edge subsets $V' \subseteq V$ and $E' \subseteq E$, we use $G(V')$ and $G(E')$ for their induced subgraphs.

For notation simplicity, we do not differ the vertex/edge set and their induced subgraphs whenever such distinction is inessential. For any $v \in V$, we use $\deg(v)$ to denote the degree of v, i.e., the number of edges incident with v. Finally, we define $\mathrm{cl}(G)$ as the closure of G obtained by repeatedly adding a new edge $e = (u, v) \in E$ for nonadjacent vertices u and v with $\deg(u) + \deg(v) \geq n$ until no more such edges can be found.

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\Pi_{\mathrm{LP}}: \quad \min_{x \in \{0,1\}^n} \ \langle -c, x \rangle$$

$$\text{s.t.} \sum_{i \in T} x_i \leq |T| - 1, \ \forall T \in m(\overline{\Omega}).$$

By definition, we have,

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

We call the latter s-t cocuts. To separate elements from $\overline{\Omega}$, we can either use the universal method provided by Algorithm 3 where the membership oracle can be implemented as a s-t subpath checking procedure, or we can heuristically generate claws (star with three edges) and cycles as they will never be a subset of any s-t simple paths. Since Ω is a lower system, generating all elimination inequalities for these cocuts is an exact solution approach for this longest simple path problem. Hence, Algorithm 3 provides a new cut generation method for solving this problem with only the elimination type of inequalities, which is quite different from other proposed formulations in the literature [18]. $\triangle$

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

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

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

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

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

- *$G(T)$ is disconnected;*
- *for some $v \in V(G(T))$, $deg(v) = 1$;*
- *there exists a vertex bipartition such that a non-singleton part has at most one vertex connected to the other part;*

- *$G(T) = cl(G(T'))$ for any above T'.*

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

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

Proposition 4. *Given $G = (V, E)$, $T \in \overline{\downarrow \Omega}$ if and only if T satisfies any of the following conditions,*

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

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

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

Through the set system approximation lens, it is somewhat counterintuitive that traditional formulations for TSP (i.e., shortest HC problem) prefer elimination inequalities over covering inequalities. Indeed, given that the objective function increases with x, Ω could be equated to its upper closure $\uparrow \Omega$, mirroring our approach in the shortest path example. This would make covering inequalities appear more naturally for derivation. However, Proposition 3 highlights the challenge in defining the structures within $\overline{\uparrow \Omega}$. As a result, classic TSP formulations focus on describing Ω by excluding all non-maximal structures from $\downarrow \Omega$. In essence, they employ elimination inequalities for non-Hamiltonian cycles (NHCs). Then, the additional constraints that ensure each vertex has a degree of two are used for two purposes: (i) eliminate the second and third types of structures, and (ii) remove every strict subset of any HC from the solution space. Note that the NHC elimination along with (i) exactly characterizes $\downarrow \Omega$ by Proposition 4, and (ii) removes those non-maximal structures in $\downarrow \Omega$.

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

4 Method Extension

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

4.1 Formulation Standardization

We will introduce two standardization techniques that could transform the original problem representation (1) to a more convenient form for applying the reformulation methods in Theorem 3. The main idea is to tailor the original set system Ω to an upper (or lower) set without affecting the optimal solutions.

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

Definition 6 (Optimality-Invariant Standardization I)**.** Given Problem (1), we perform two steps to standardize the solution space $\mathcal{X}$ before applying the approximation reformulations.

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

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

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

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

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

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

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

to obtain the optimal solution that represents T.

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

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

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

Definition 7 (Entry-wise Monotone Functions)**.** Given a function $f : \{0,1\}^n \to \mathbb{R}$, let (k, x_{-i}) denote a vector from $\{0,1\}^n$ where the ith entry is equal to $k \in \{0,1\}$ and the rest entries are equal to $x_{-i} \in \{0,1\}^{n-1}$, and define

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

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

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

Proposition 5. *The following functions are entry-wise monotone:*

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

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

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

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

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

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

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

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

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

Proposition 6 (Optimality-Invariant Standardization II)**.** *Given Problem* (1) *with an entry-wise monotone objective function, we can obtain an equivalent reformulation with an increasing (or decreasing) function by setting $x_i = 1 - x'_i$ for all $i \in I := \{i \in \Delta \mid \Delta_{x_{-i}} < 0\}$ (or $I := \{i \in \Delta \mid \Delta_{x_{-i}} > 0\}$).*

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

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

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

With both standardization methods, we are able to transform a larger class of problems (1) to have upper or lower systems as solution spaces, so that the reformulations in Theorem 3 become exact. For cases where such standardization methods do not apply, we introduce a set system decomposition methodology in the next subsection.

4.2 Set System Decomposition

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

Definition 8 (Interval Set System)**.** A set system Ω is said to have the interval property if for every $T, T' \in \Omega$, any T'' sandwiched between T and T' is also in Ω.

Lemma 3. *A set system Ω has the interval property if and only if it can be written as $\Omega = \Omega_1 \cap \Omega_2$ for some upper system Ω_1 and some lower system Ω_2.*

Proof. For necessity, we show that every interval set system Ω can be written as $\Omega = (\uparrow \Omega) \cap (\downarrow \Omega)$. Take any $T \in \Omega$, we have $T \in \uparrow \Omega$ as well as $T \in \downarrow \Omega$, which proves one inclusion. For the other direction, take any $T \in (\uparrow \Omega) \cap (\downarrow \Omega)$. By the definitions of upper and lower closures, there exist

$T', T'' \in \Omega$ such that $T' \subseteq T \subseteq T''$. Then, the interval property of Ω ensures that $T \in \Omega$. For sufficiency, take any $T, T' \in \Omega$ such that $T \subseteq T'$. We have $T \in \Omega_1$ and $T' \in \Omega_2$. Take any T'' sandwiched between T and T', the upward closure of Ω_1 and the downward closure of Ω_2 enforce that $T'' \in \Omega_1 \cap \Omega_2 = \Omega$, which shows that Ω is an interval set system. $\square$

This lemma indicates that the following reformulation is exact for any interval system Ω.

Corollary 6. *Suppose the solution space Ω of Problem (1) is an interval set system, then the following is an exact reformulation of (1).*

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

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

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

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

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

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

Theorem 6 (Set System Decomposition Theorem)**.** *Every set system Ω can be decomposed as $\Omega = \bigcup_{k \in K} \Omega_k$ for a finite family of interval set systems $\{\Omega_k\}_{k \in K}$.*

Proof. We provide a rather explicit decomposition as follows,

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

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

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

Corollary 7. *Given any set system* Ω *with a decomposition* $\Omega = \bigcup_{k \in K} \Omega_k$ *for some interval set systems* $\{\Omega_k\}_{k \in K}$*, problem* (1) *can be equivalently reformulated as*

$$\min_{x \in \{0,1\}^n, z \in \{0,1\}^{|K|}} \; f(x) \tag{8a}$$

$$\text{s.t.} \; \sum_{i \in T} x_i \geq z_k, \qquad \forall k \in K, \forall T \in m\left(\widehat{\uparrow \Omega_k}\right), \tag{8b}$$

$$\sum_{i \in S} x_i \leq |S| - z_k, \; \forall k \in K, \forall S \in m\left(\overline{\downarrow \Omega_k}\right), \tag{8c}$$

$$\sum_{k \in K} z_k = 1. \tag{8d}$$

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

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

4.3 Examples

In this subsection, we provide two examples to explore the potential applications of the proposed standardization techniques and set system decomposition method in multistage and stochastic optimization settings.

Example 6 (Multistage Problems)**.** Many multistage problems can be reformulated into the following problem,

$$\min_{x \in \mathcal{X}_\Omega} \min_{y \in \mathcal{Y}_x} (\text{or} \max) \; f(x, y) \tag{9a}$$

where x and y are binary and continuous, respectively. Suppose an interval set decomposition can be identified for Ω or $\theta(x) := \min_{y \in \mathcal{Y}_x} f(x, y)$ can be standardized into a monotone function on x, then the corresponding proposed reformulations can be applied to induce a cut generation implementation to describe the first stage solution space $\mathcal{X}_\Omega$. △

Example 7 (BIPs with Distributionally Robust Chance Constraints)**.** Consider the following BIP with a set of chance constraints,

$$\min_{x \in \mathcal{X}_\Omega} \; f(x) \tag{10a}$$

$$\text{s.t.} \; \min_{\mathbb{P} \in \mathfrak{P}} \; \mathbb{P}(\langle a_i(\xi), x \rangle \geq b(\xi)) \geq 1 - \epsilon, \; \forall i \in I, \tag{10b}$$

where $a_i(\xi) \geq 0$, and ξ is an uncertain vector following the probability law $\mathbb{P}$, which is the worst among the set $\mathfrak{P}$. We further assume two things: (i) Ω is an upper system; (ii) for a given x, the evaluation of (10b) can be done efficiently using a verifier $\mathbb{I}_\epsilon$. Then, the entire solution space is associated with the following set system

$$\Omega_\epsilon := \Omega \cap \Omega'_\epsilon, \text{ where } \Omega'_\epsilon := \{T_x \subseteq \Delta \mid x \text{ satisfies (10b)}\}.$$

Note that Ω'_ϵ is also an upper system due to the nonnegativity of a_i's, which is independent of the definition of the ambiguity set $\mathfrak{P}$. By the main theorem, this problem can be equivalently reformulated into (3) with the index set $\widehat{\overline{\Omega_\epsilon}}$. To separate such constraint, we can use Algorithm 1 with the oracle $\mathbb{I}_{\Omega_\epsilon} = \mathbb{I}_\Omega \cdot \mathbb{I}_\epsilon$ where the dot is the multiplication. Both verification oracles are assumed to be efficient, and so is the separation procedure.

Now, consider the same problem with the following variant of (10b)

$$\min_{\mathbb{P} \in \mathfrak{P}} \mathbb{P}(\langle a_i(\xi), x\rangle \leq b(\xi)) \geq 1 - \epsilon, \ \forall i \in I,$$

where the inner constraints become less than or equal to. Then, Ω'_ϵ becomes a lower system, which implies that Ω_ϵ is an interval set. We can instead use Formulation (7) with index sets $\widehat{\widehat{\Omega}}$ and $\overline{\Omega'_\epsilon}$ for (7b) and (7c), which can be separated using Algorithms 1 and 3, respectively. △

5 Structural Duality

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

Definition 9 (Structural Duality)**.** Given any upper (lower) set system $\Omega \subseteq \Delta$, we call $\widehat{\widehat{\Omega}}$ the upper (lower) dual system of Ω. We also call the pair of dual structures $\left(\Omega, \widehat{\widehat{\Omega}}\right)$ a *cut* (*cocut*) pair. Moreover, for an arbitrary set system Ω, we call $\widehat{\widehat{\uparrow \Omega}}$ and $\widehat{\widehat{\downarrow \Omega}}$ the cut and cocut completions, respectively.

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

Proposition 7. *A cut (cocut) pair* $\left(\Omega, \widehat{\widehat{\Omega}}\right)$ *satisfies the following properties:*

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

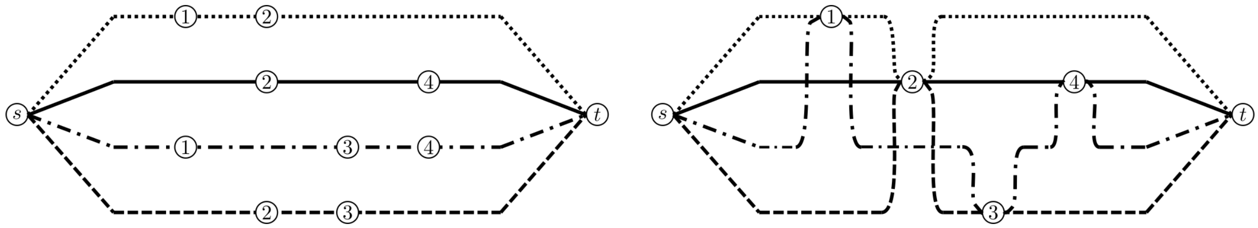

Figure 2: General Max-Flow Min-Cut Theorem. Given $m(\Omega) = \{\{1,2\},\{2,3\},\{2,4\},\{1,3,4\}\}$, we first construct one pipe for each $T \in m(\Omega)$, and connects these pipes with two artificial nodes s and t (left figure). Then, we merge nodes on different pipes and assign c_i as the capacities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

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

Theorem 7 (General Min-Cut Max-Flow Duality)**.** *For any cut pair Ω and $\widehat{\widehat{\Omega}}$ with a non-negative cost vector c assign to the ground set Δ, there exists a flow function f with c as the capacity vector such that*

$$\min_{x \in \mathcal{X}_{\widehat{\widehat{\Omega}}}} \langle c, x \rangle = \max_{x \in \mathcal{X}_{\Omega}} f(c, x), \tag{11}$$

and f is increasing on both c and x.

Proof. We have $\widehat{\widehat{\Omega}} = \mathcal{C}(\Omega) = \mathcal{C}(m(\Omega))$. Hence, the left side of (11) equals

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

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

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

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

By this design, the function f is clearly increasing on c and x since a larger c provides more capacity on the checkpoints, and a larger x implies more checkpoints are open for flow to pass. Since Ω is an upper system by the definition of the cut pair, the right side of (11) equals $f(c, 1)$, i.e., the maximum flow in the constructed graph with capacities c and with all the checkpoints open. Moreover, this constructed graph ensures that every cut in $\mathcal{C}(\Omega)$ corresponds to a unique plan to disconnect all the pipes by blocking the corresponding checkpoints. Moreover, the maximum combined flow in the graph equals the bottleneck among all such cuts, which equals the value of min-cut. This concludes the proof. $\square$

The reason we designed a max-flow problem with separated pipes is to prevent the possibility that shared pipes may create extra paths for increasing the total flow (see Figure 2). We also note that this max-flow problem is impractical to implement as it requires knowing all the minimal structures in $m(\Omega)$ to construct the graph. Instead, this theorem mainly serves as a conceptual vehicle to generalize the classic max-flow min-cut theorem to all the binary optimization problems whenever $\Omega_{\mathcal{X}}$ is an upper or lower system. This further confirms that the structural dual pair defined in Definition 9 is indeed a generalization of the classic duality relationship between the s-t connected subgraphs and s-t cuts.

5.1 Dual Pairs of Common Structures

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

Example 8 (Spanning Trees)**.** Given a graph $G = (V, E)$, a spanning tree is a connected subgraph without cycles. Let Ω be the set of spanning trees, then $\uparrow \Omega$ and $\downarrow \Omega$ are the set of connected subgraphs and the set of forests, respectively. A direct computation shows that

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

We call the former the edge cuts and the latter the co-cycles. Hence, connected subgraphs and edge cuts form a cut pair, and the forests and co-cycles form a cocut pair. $\triangle$

Example 9 (Knapsack (Set Packing) Problem)**.** The classic knapsack problem aims to identify an item list that is below the knapsack's capacity w to optimize the given objective function. Hence, the solution space Ω_w consists of all item lists that are under the given capacity w, which is clearly a lower system. Let W be the total item weight and $\epsilon > 0$ be a sufficiently small value, a direct computation shows that the corresponding dual structure is the following,

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

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

Example 10 (Vertex Covers & Independent Sets)**.** Given a graph $G = (V, E)$, a vertex cover is a vertex subset $T \subseteq V$ that includes at least one endpoint of every edge of the graph. The system Ω of vertex covers is clearly an upper system. Then, a direct computation shows the dual structures is the following,

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

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

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

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

Example 11 (Mathchings)**.** Given a graph $G = (V, E)$, a feasible matching is a set of edges $T \subseteq E$ that have mutually disjoint vertices. Hence, the system of matchings Ω is lower closed. Then, a direct computation shows that its cocut counterpart is the following.

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

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

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

6 Conclusion

This paper presents a generic reformulation technique capable of transforming any given BIP into a format involving only set covering and subtour elimination inequalities, along with a singular choice constraint. This wide applicability is achieved through the development of set system approximation theory, grounded in the intriguing principles of cut-cocut algebra. This fresh analytical perspective not only facilitates a new reformulation technique but also connects advancements in set covering and elimination inequalities to arbitrary BIPs. This could lead to innovative solution strategies for various instance problems. Additionally, these developments unveil a set system duality concept that resonates with the classic max-flow min-cut theorem.

This approach, rooted in set system approximation, opens avenues for numerous interesting research questions. One area of inquiry explores whether set operators corresponding to other types of inequalities, like packing and partition constraints, exist. If they do, their algebraic interplay with other operators and their utility in BIP reformulation warrant investigation. Another intriguing aspect is the potential existence of an efficient set system decomposition algorithm for specific system types. Moreover, the connection between these reformulations and the solution convex hull merits further study. For instance, in certain interdiction games, covering inequalities are often found to be facet-defining and are more potent than Benders decomposition inequalities. This raises the question: Can we extend this observation more broadly? Furthermore, in our examples, we noted a link between the complexity of a BIP and the effectiveness in characterizing its upper and lower dual structures, hinting at a possible complexity characterization related to these dual structures.

Numerous methodologies for analyzing BIPs have emerged in the literature, including geometric approaches from polyhedron theory, network insights from graph theory, and coefficient manipulation strategies from linear algebra. These diverse perspectives create a rich field for exploration. Our proposed framework adds a new toolset for examining the algebraic properties of set systems and their operators, potentially spurring additional research questions and enriching the field further.

References

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

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

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

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

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

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

[7] M. Conforti, G. Cornuéjols, and G. Zambelli. Extended formulations in combinatorial optimization. *4OR*, 8(1):1–48, 2010.

[8] G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8(1):101–111, 1960.

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

[10] J. Farkas. Theorie der einfachen ungleichungen. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, 1902(124):1–27, 1902.

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

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

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

[14] C. N. Jones, E. C. Kerrigan, and J. M. Maciejowski. On polyhedral projection and parametric programming. *Journal of Optimization Theory and Applications*, 138:207–220, 2008.

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

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

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

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

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

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

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

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

[23] F. Vanderbeck and L. A. Wolsey. *Reformulation and decomposition of integer programs*. Springer, 2010.

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

[25] N. Wei and J. L. Walteros. On supervalid inequalities for binary interdiction games. 2024.

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

[27] L. A. Wolsey. Generalized dynamic programming methods in integer programming. *Mathematical Programming*, 4(1):222–232, 1973.

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