Modeling Binary Relations in Piecewise-Linear Approximations*

Kristin Braun[1] Robert Burlacu[2] Tobias Kuen[1,*] Kristina Rolsing[2]

[1]Gruppe Optimization, Fraunhofer-Institut für Integrierte Schaltungen IIS, Nordostpark 84, 90411 Nürnberg, Germany

[2]University of Technology Nuremberg, 90411 Nürnberg, Germany

*Corresponding author: tobias.kuen@iis.fraunhofer.de

Abstract

Over the last decades, using piecewise-linear mixed-integer relaxations of nonlinear expressions has become a strong alternative to spatial branching for solving mixed-integer nonlinear programs. Since these relaxations give rise to large numbers of binary variables that encode interval selections, strengthening them is crucial. We investigate how to exploit the resulting combinatorial structure by integrating cutting-plane techniques directly in the binary variable space. At the core lies a multipartite generalization of the bipartite implication polytope, capturing conditional relations between several groups of multiple-choice variables and an implied selection. We analyze the polyhedral structure of this polytope and a natural set-valued variant, deriving a unifying family of valid inequalities and characterizing all nontrivial facets. Building on this theory, we design a generic separation algorithm and embed it into standard multiple-choice and incremental piecewise-linear mixed-integer relaxations of mixed-integer nonlinear programs. We further exploit structure in the underlying nonlinearities to precompute strong cuts for relevant application classes such as pooling problems and Gaussian processes. Extensive computational experiments on MINLPLib instances demonstrate that the proposed cuts significantly tighten the linear relaxations and thereby reduce solution times.

Keywords: Piecewise-Linear Approximations, Mixed-Integer Nonlinear Programming, Cutting Planes, Boolean Quadric Polytope, Bipartite Implication Polytope, Multipartite Implication Polytope

1 Introduction

Mixed-integer nonlinear programs (MINLPs) are pivotal in modeling complex systems across various disciplines. They combine discrete and continuous variables and involve nonlinear (nonconvex) relationships, which makes them challenging to solve in practice. A prevalent strategy to address this difficulty is to employ piecewise-linear (PWL) relaxations of nonlinear functions inside a disjunctive programming approach. This transforms the original MINLP into a mixed-integer linear program (MIP) that can be treated with state-of-the-art solvers. The appeal of this methodology is that it builds on mature branch-and-cut technology and yields a solution framework that is often much more tractable than directly branching on nonlinear variables. A central aspect for the practical performance of such MIP-based approaches is the strength of the continuous relaxation. Weak relaxations typically lead to large branch-and-bound trees and long solution times.

*This work was part of the project EKSSE, funded by the Federal Ministry of Education and Research (BMBF).

Consequently, recent advances have to a large extent focused on tightening linear relaxations, both by improved formulation techniques and by algorithmic enhancements in the solution process; see for instance [4, 10, 14, 15]. Strong and compact formulations for piecewise-linear models have been studied extensively; we refer to the survey in [18] and to the unifying framework in [19], as well as to recent advances for nonconvex PWL functions in [6].

The most widely used cuts, such as McCormick envelopes [12], primarily act on continuous variables to tighten convex relaxations of nonlinear terms. In the context of separable nonconvex piecewise-linear objective functions, families of valid inequalities have been derived and embedded into branch-and-cut frameworks [8]. This line of work has been extended to lower semicontinuous PWL functions [20], and a detailed polyhedral analysis of the associated piecewise-linear optimization polytope together with further facet-defining inequalities is provided in [22]. Beyond PWL objectives, polyhedral strengthening has also been studied for structured PWL constraint systems. In [13], the piecewise-linearized-flow polytope arising in network optimization is investigated and valid inequalities are derived. On a broader MINLP level, disjunctive cutting planes derived from convex disjunctions have been proposed to strengthen nonlinear relaxations in branch-and-cut frameworks; see, for instance, [16, 2]. To the best of our knowledge, however, the combinatorial structure induced by interval-selection variables in general PWL relaxations has not yet been systematically exploited within a dedicated cutting-plane framework.

Such relaxations introduce binary variables that encode interval selections and satisfy structured dependencies induced by the discretized nonlinear expression. This creates the opportunity to leverage these relations directly in the binary variable space, whereas generic MIP cuts are typically not tailored to this PWL-induced structure. A notable polyhedral structure that encapsulates these relations is the bipartite implication polytope (BIP) from [5], which models conditional relations across three sets of binary variables. This polytope captures such conditional relationships and forms the basis for deriving valid inequalities that strengthen the linear relaxation. In this work, we extend this theory from relations across two implying sets to relations across arbitrarily many sets and term the resulting structure the multipartite implication polytope (MPIP).

Contribution. The main contribution of this work is to show that the binary structures induced by piecewise-linear relaxations of nonlinear expressions can be systematically exploited at the level of the interval-selection variables via implication polytopes. Building on the bipartite implication polytope introduced in [5], we show that this structure arises not only in isolated nonlinear constraints, but also as a recurring substructure in expression-tree based reformulations of general MINLPs. We extend this framework to the multipartite implication polytope, allowing for an arbitrary number of implying sets, and to a set-valued variant that models cases in which several implied indices are admissible.

For these polytopes, we derive a unifying family of valid inequalities, the general MPIP inequalities, and prove that, up to trivial bounds, every facet is induced by an inequality of this type. On the algorithmic side, we develop a generic separation routine for general MPIP inequalities that operates directly in the binary variable space and can be integrated into standard multiple-choice [7] and incremental piecewise-linear relaxation formulations [11]. In addition, we exploit special structure in the underlying relation mappings of selected application classes to precompute strong problem-specific cuts, such as corner and stripe inequalities, which can be added to the formulation a priori.

A computational study on a broad subset of MINLPLib instances, including pooling and kriging peaks problems, demonstrates that these enhancements significantly tighten the linear relaxations and lead to substantial reductions in solution time, while rendering multiple-choice formulations competitive with incremental formulations.

Structure of the Paper The remainder of the paper is organized as follows. Section 2 recalls the definition and polyhedral description of the bipartite implication polytope and explains how this structure arises in piecewise-linear relaxations, both for elementary operators and along expression trees. Section 3 introduces the multipartite implication polytope and its set-valued extension, develops the corresponding polyhedral theory, and characterizes the facets via the class of general MPIP inequalities. Section 4 shows how these inequalities can be embedded into different piecewise-linear MIP reformulations of general nonlinear constraints. Section 5 focuses on application-specific structure and describes how to precompute families of corner and stripe inequalities for pooling and kriging peaks instances. Section 6 reports the results of a computational study on MINLPLib, comparing standard and MPIP-enhanced formulations. We conclude with a summary and directions for future research.

2 The Bipartite Implication Polytope and its Connection to Piecewise-Linear Approximations

In this section, we recall the definition and polyhedral description of the bipartite implication polytope (BIP) introduced in [5] and relate it to standard piecewise-linear (PWL) relaxations of nonlinear constraints. We first summarize the essential structural properties of the BIP and the associated family of valid inequalities. We then show how BIP structures naturally arise in PWL formulations, both for isolated operators such as bilinear terms and, more generally, along expression-tree based reformulations.

2.1 Introduction to the Bipartite Implication Polytope

The BIP captures conditional relations across three sets of binary variables: choosing one variable in each of two *implying* sets enforces the choice of exactly one variable in an *implied* set. This situation arises naturally, for example, when products of binary variables are grouped by identical costs or when scenario variables are coupled to underlying decision variables. The BIP was introduced and analyzed in detail in [5], where a complete polyhedral description was derived.

2.1.1 Definition

Let $\alpha, \omega, \beta \in \mathbb{N}$ and let

$$\mathbf{x}^1 \in \{0,1\}^\alpha, \qquad \mathbf{x}^2 \in \{0,1\}^\omega, \qquad \mathbf{y} \in \{0,1\}^\beta$$

be three vectors of binary variables, subject to multiple-choice constraints in each block. The implication pattern is encoded by a mapping

$$\varphi : [\alpha] \times [\omega] \to [\beta],$$

where $\varphi(j_1, j_2)$ specifies which y–variable must be activated if $x^1_{j_1} = 1$ and $x^2_{j_2} = 1$. The set of feasible 0–1 points is therefore

$$P(\varphi) := \operatorname{conv}\Big\{(\mathbf{x}^1, \mathbf{x}^2, \mathbf{y}) \in \{0,1\}^{\alpha+\omega+\beta} \,:\, x^1_{j_1} x^2_{j_2} \le y_{\varphi(j_1,j_2)} \ \forall (j_1, j_2) \in [\alpha] \times [\omega],$$
$$\sum_{j_1=1}^{\alpha} x^1_{j_1} = \sum_{j_2=1}^{\omega} x^2_{j_2} = \sum_{l=1}^{\beta} y_l = 1\Big\}.$$

For the remainder of this paper we assume $\alpha = \omega$ since it simplifies notation.

2.1.2 Convex Hull Description

In [5], a new family of valid inequalities, called *n-block inequalities*, was introduced for the BIP and shown to provide, together with the trivial bounds, a complete linear description of $P(\varphi)$. In the notation used here, these inequalities have the form

$$\sum_{j_1 \in [\alpha]} a^1_{j_1} x^1_{j_1} + \sum_{j_2 \in [\alpha]} a^2_{j_2} x^2_{j_2} \leq \sum_{l \in [\beta]} b_l y_l + n,$$

where the respective variable coefficients are given by

$$\begin{aligned} a^1_{j_1} &= |\{k \in [n] \,:\, j_1 \in X^1_k\}|, \quad j_1 \in [\alpha], \\ a^2_{j_2} &= |\{k \in [n] \,:\, j_2 \in X^2_k\}|, \quad j_2 \in [\alpha], \\ b_l &= \max\{k \in [n] \,:\, l \in \Xi^M_k([n])\}, \quad l \in [\beta]. \end{aligned}$$

Here, the sets $X^1_k \subseteq [\alpha]$ and $X^2_k \subseteq [\alpha]$ define a nested family of blocks in the implication structure, and the sets $\Xi^M_k([n])$ encode which y–indices arise in the intersection of at least k blocks.

The central structural result for the BIP in [5] is the following.

Theorem 1 (Convex hull description of the BIP [5, Theorem 4.2])**.** *The full convex-hull description of $P(\varphi)$ is given by the multiple-choice constraints, the non-negativity constraints and the n-block constraints for $n \leq \bar{n}$ for some fixed $\bar{n} \in \mathbb{N}$.*

2.2 Introduction to MINLPs and Piecewise-Linear Relaxations

We consider mixed-integer nonlinear programs (MINLPs) with linear objective functions and linear as well as nonlinear constraints. This does not reduce generality for our purposes, as a nonlinear objective function can be reformulated by introducing an auxiliary variable and an additional nonlinear constraint. The variables are mixed-integer, i.e., each component of the variable vector is either real-valued or integer-valued. The formulation of such an MINLP with m linear constraints, k nonlinear constraints, q continuous variables and p integer variables is given as follows.

$$\min_{\boldsymbol{\chi}} \ \mathbf{c}^\top \boldsymbol{\chi} \tag{1a}$$

$$\text{s.t. } \mathbf{A}\boldsymbol{\chi} \leq \mathbf{b}, \tag{1b}$$

$$f(\boldsymbol{\chi}) \leq 0 \quad \text{for all } f \in \mathcal{F}, \tag{1c}$$

$$\boldsymbol{\chi}_L \leq \boldsymbol{\chi} \leq \boldsymbol{\chi}_U, \tag{1d}$$

$$\boldsymbol{\chi} \in \mathbb{R}^q \times \mathbb{Z}^p. \tag{1e}$$

The objective function in (1a) is defined by the linear objective vector $\mathbf{c} \in \mathbb{R}^{q+p}$. In (1b), the linear constraints are given using $\mathbf{A} \in \mathbb{R}^{m \times (q+p)}$ and $\mathbf{b} \in \mathbb{R}^m$. Let $\mathcal{F}$ be a finite set of nonlinear functions. Then, all nonlinearities are represented in Constraint (1c) by nonlinear real-valued functions $f : D_f \to \mathbb{R}$ with domains $D_f \subset \mathbb{R}^{q+p}$. The variable vector $\boldsymbol{\chi}$ might be bounded from below and above by $\boldsymbol{\chi}_L \in (\mathbb{R} \cup \{-\infty\})^{q+p}$ and $\boldsymbol{\chi}_U \in (\mathbb{R} \cup \{\infty\})^{q+p}$, see (1d).

To overcome the complex nature of MINLPs, piecewise-linear (PWL) approximation approaches aim to reformulate MINLPs as MIPs and exploit the performance of modern MIP solvers. This approach comes with one restriction: Any variable χ^i that appears in nonlinear expressions must be bounded by some $\chi^i_L, \chi^i_U \in \mathbb{R}$. The main idea of PWL approximations is to partition the domain of a function into multiple segments and to determine in which segment a point lies using a set of breakpoints. When n segments are introduced, $n+1$ breakpoints $\eta_0, \eta_1, \ldots, \eta_n$ are needed. In this work, we assume all nonlinearities to be one-dimensional, i.e., we only have functions of the form $f : \mathbb{R} \to \mathbb{R}$, or bilinear, i.e., $f : \mathbb{R}^2 \to \mathbb{R}$ with $f(\chi^1, \chi^2) = \chi^1 \chi^2$.

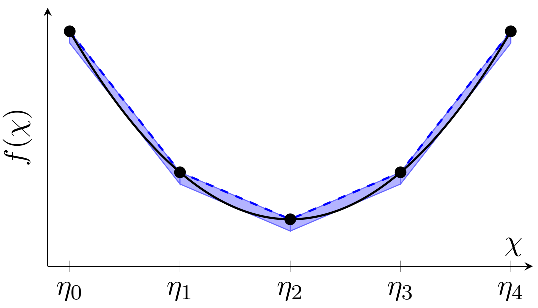


Figure 1: Nonlinear function $f(\chi) = \chi^2$ (black) with approximation (dashed blue) and relaxation (transparent blue). The breakpoints η are given as black circles.

By introducing auxiliary variables, any nonlinear expression can be decomposed into elementary univariate or bilinear operations with the use of expression trees that are explained later on in Section 2.3.2.

Every approximation introduces approximation errors. Thus, the obtained solution is only feasible up to some tolerance ϵ. The graphs coincide at the breakpoints; between breakpoints the approximation deviates by at most ϵ. To ensure that the full feasible set of the MINLP is contained in the feasible set of the reformulated problem, approximations can be extended to relaxations by including a surrounding region around the approximation.

A graphical example of a nonlinear function, and a possible approximation and relaxation with 4 segments and 5 breakpoints is given in Figure 1.

PWL relaxations can be described with MIPs. For the linearization, each nonlinear expression

$$f(\chi) = z$$

is replaced by additional constraints and artificial variables. The number of variables and constraints depends on the model that is used. In [3], the most common models for reformulating MINLPs using PWL are presented and compared. One of the state-of-the-art models is the incremental method – introduced by [11] back in 1957. It is defined by

$$\eta_0 + \sum_{i=1}^{n} \delta_i(\eta_i - \eta_{i-1}) = \chi, \tag{2a}$$

$$f(\eta_0) + \sum_{i=1}^{n} \delta_i(f(\eta_i) - f(\eta_{i-1})) = z, \tag{2b}$$

$$\delta_1 \leq 1, \tag{2c}$$

$$\delta_{i+1} \leq x_i \qquad \text{for all } i \in [n-1], \tag{2d}$$

$$x_i \leq \delta_i \qquad \text{for all } i \in [n], \tag{2e}$$

$$\delta_n \geq 0, \tag{2f}$$

$$x_i \in \{0, 1\} \qquad \text{for all } i \in [n]. \tag{2g}$$

Another model is given by the multiple-choice method:

$$\sum_{i=1}^{n} \hat{\chi}_i = \chi, \tag{3a}$$

$$\sum_{i=1}^{n} (m_i \hat{\chi}_i + t_i x_i) = z, \tag{3b}$$

$$\sum_{i=1}^{n} x_i = 1, \tag{3c}$$

$$x_i \eta_{i-1} \leq \hat{\chi}_i \qquad \text{for all } i \in [n], \tag{3d}$$

$$\hat{\chi}_i \leq x_i \eta_i \qquad \text{for all } i \in [n], \tag{3e}$$

$$x_i \in \{0, 1\} \qquad \text{for all } i \in [n]. \tag{3f}$$

In this formulation, each part of the piecewise-linearity is represented by a linear equation $m_i \hat{\chi}_i + t_i$.

Many other models rely on the convex combination method [19, 6]. As we are not using these models in this work, we do not go into further detail here.

These MIP models have in common that active segments are modeled using auxiliary binary variables. The exact number and usage of binary variables depends on the formulation. Depending on that, we can apply our theory there, see Section 4 for more details.

2.3 Appearance of Bipartite Implication Polytope Structures in MINLPs

The piecewise-linear relaxations introduced in the previous subsection give rise to auxiliary binary variables that encode interval membership of the original continuous variables. Whenever these binary variables are coupled through discretized images of multivariate operators (such as bilinear terms), the resulting combinatorial structure is naturally described by instances of the BIP. In this section we formalize this connection for bilinear terms under a piecewise constant relaxation and illustrate it on a small example. We then explain how similar structures arise systematically when expression trees are used to reformulate general nonlinear constraints.

2.3.1 Piecewise Constant Relaxation of Bilinear Terms

Consider a bilinear constraint

$$\xi = \chi^1 \chi^2,$$

where $\chi^1 \in [\chi^1_L, \chi^1_U]$, $\chi^2 \in [\chi^2_L, \chi^2_U]$, and $\xi \in [\xi_L, \xi_U]$. Let

$$\begin{aligned} &\chi^1_L = \eta^1_0 < \eta^1_1 < \cdots < \eta^1_\alpha = \chi^1_U, \\ &\chi^2_L = \eta^2_0 < \eta^2_1 < \cdots < \eta^2_\alpha = \chi^2_U, \text{ and,} \\ &\xi_L = \tau_0 < \tau_1 < \cdots < \tau_\beta = \xi_U \end{aligned}$$

be discretizations of the domains of χ^1, χ^2, and ξ, respectively. Define intervals

$$I^1_{j_1} = [\eta^1_{j_1-1}, \eta^1_{j_1}),\ j_1 \in [\alpha], \qquad I^2_{j_2} = [\eta^2_{j_2-1}, \eta^2_{j_2}),\ j_2 \in [\alpha], \qquad J_l = [\tau_{l-1}, \tau_l),\ l \in [\beta].$$

Introduce binary variables

$$x^1_{j_1} \in \{0,1\},\ j_1 \in [\alpha], \qquad x^2_{j_2} \in \{0,1\},\ j_2 \in [\alpha], \qquad y_l \in \{0,1\},\ l \in [\beta],$$

encoding the selected intervals of χ^1, χ^2, and ξ. A standard multiple-choice formulation imposes

$$\sum_{j_1=1}^{\alpha} x^1_{j_1} = 1, \qquad \sum_{j_2=1}^{\alpha} x^2_{j_2} = 1, \qquad \sum_{l=1}^{\beta} y_l = 1,$$

and links the continuous variables to the discrete choices by suitable linear constraints ensuring

$$x^1_{j_1} = 1 \;\Rightarrow\; \chi^1 \in I^1_{j_1}, \quad x^2_{j_2} = 1 \;\Rightarrow\; \chi^2 \in I^2_{j_2}, \quad y_l = 1 \;\Rightarrow\; \xi \in J_l.$$

The remaining task is to model the graph of the bilinear mapping $\xi = \chi^1\chi^2$ at the level of interval selections. To this end, define a set-valued mapping

$$\Phi : [\alpha] \times [\alpha] \to 2^{[\beta]}$$

by

$$\Phi(j_1, j_2) \coloneqq \Big\{ l \in [\beta] \,\Big|\, \{\chi^1\chi^2 : \chi^1 \in I^1_{j_1},\ \chi^2 \in I^2_{j_2}\} \cap J_l \neq \emptyset \Big\}.$$

A piecewise constant relaxation of the bilinear term at the level of interval indices enforces the logical implication

$$x^1_{j_1} = 1,\ x^2_{j_2} = 1 \;\Rightarrow\; y_l = 1 \text{ for some } l \in \Phi(j_1, j_2),$$

i.e., whenever χ^1 and χ^2 fall into intervals $(I^1_{j_1}, I^2_{j_2})$, the variable ξ must lie in one of the intervals J_l with index $l \in \Phi(j_1, j_2)$. This bipartite implication structure is similar to a BIP instance with the difference that the relation function Φ is set-valued. We will address the extension of BIP theory to set-valued relation functions later in Section 3.2. In this sense, every bilinear term discretized by a piecewise constant relaxation gives rise to an embedded structure similar to the BIP in the mixed-integer relaxation.

		x^2			
	Φ	1	2	3	4
x^1	1	{1}	{1}	{1}	{1}
	2	{1}	{1}	{1,2}	{1,2}
	3	{1}	{1,2}	{2,3}	{2,3}
	4	{1}	{1,2}	{2,3}	{3,4}

Figure 2: Set-valued mapping Φ for the bilinear expression $\xi = \chi^1\chi^2$. Cell (j_1, j_2) displays $\Phi(j_1, j_2)$.

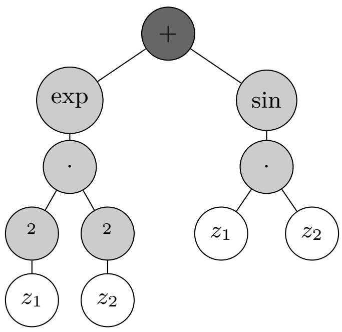


Figure 3: Expression Tree for $\exp((z_1)^2 \cdot (z_2)^2) + \sin(z_1 z_2)$

Example 1. *Assume bounded domains* $\chi^1, \chi^2 \in [0,4)$, $\xi \in [0,16)$. *Discretize each domain into* 4 *intervals:*

$$\begin{aligned} I^1_{j_1} &= [j_1 - 1, j_1) && \text{for } \chi^1, \\ I^2_{j_2} &= [j_2 - 1, j_2) && \text{for } \chi^2, \\ J_l &= [4(l-1), 4l) && \text{for } \xi, \end{aligned}$$

where $j_1, j_2, l \in [4]$. *Binary variables* $x^1_{j_1}, x^2_{j_2}, y_l$ *indicate membership in* $I^1_{j_1}, I^2_{j_2}, J_l$, *respectively. The relation mapping* $\Phi\colon [4]^2 \to 2^{[4]}$ *is defined such that* $\Phi(j_1, j_2)$ *contains* l *if there exist* $\chi^1 \in I^1_{j_1}$, $\chi^2 \in I^2_{j_2}$ *with* $\chi^1\chi^2 \in J_l$. *Using the explicit bounds of the intervals, we obtain*

$$\Phi(j_1, j_2) = \left\{ l \in [4] : \left[(j_1 - 1)(j_2 - 1),\ j_1 j_2\right) \cap \left[4(l-1), 4l\right) \neq \emptyset \right\}.$$

Figure 2 visualizes Φ, *where cell* (j_1, j_2) *lists the set* $\Phi(j_1, j_2)$. *For instance:*

- $\Phi(1,1) = \{1\}$ *since* $\chi^1 \in [0,1)$, $\chi^2 \in [0,1)$ *implies* $\xi = \chi^1\chi^2 \in [0,1) \subseteq [0,4) = J_1$.
- $\Phi(2,4) = \{1,2\}$ *since* $\chi^1 \in [1,2)$, $\chi^2 \in [3,4)$ *implies* $\xi = \chi^1\chi^2 \in [3,8) = [3,4) \cup [4,8) \subseteq J_1 \cup J_2$.

(to be continued in Example 5)

2.3.2 Expression Tree

Beyond the explicit model equations, additional instances of the BIP arise from expression trees, i.e., graph-based representations of how a complex nonlinear expression is built from simpler subexpressions using basic operators. When using expression trees [9] to reformulate the problem such that it contains only one-dimensional nonlinear functions, relations between original variables and artificially created variables emerge naturally and can again be encoded through bipartite implication structures after discretization.

Example 2. *Consider the constraint*

$$\exp((z_1)^2 \cdot (z_2)^2) + \sin(z_1 z_2) = 0.$$

The corresponding expression tree is shown in Figure 3. Each gray node represents an operation applied to its children, and each leaf corresponds to an original variable. Introducing an auxiliary variable for each gray node yields the one-dimensional expressions

$$z_3 = z_4 + z_5$$

$$z_4 = \exp(z_6) \qquad z_5 = \sin(z_7)$$

$$z_6 = z_8 z_9 \qquad z_7 = z_1 z_2$$

$$z_8 = z_1^2 \qquad z_9 = z_2^2.$$

Suppose now that the variables at the root and at the leaves are discretized, e.g., by partitioning the domain of each such variable into finitely many intervals and introducing binary variables that select one interval per variable. For any subtree whose root and leaves are discretized, the intervening operations induce bipartite implication relations. The bilinear maps for example $z_6 = z_8 \cdot z_9$ *and* $z_7 = z_1 \cdot z_2$ *induce relations of the same type as in the bilinear case discussed above. In addition, we obtain the implication structure from the equation* $z_6 = (z_1)^2(z_2)^2$. *In aggregate, a single nonlinear constraint such as* $\exp((z_1)^2(z_2)^2) + \sin(z_1 z_2) = 0$ *can generate several overlapping BIP instances, all embedded in the same mixed-integer formulation.*

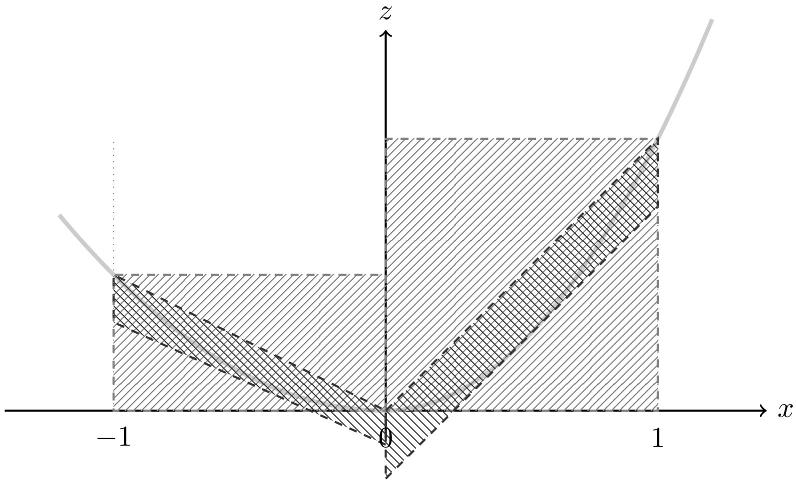


Figure 4: Univariate example for the tightening of the PWL relaxation. The standard PWL relaxation yields the parallelograms (north-west hatching). The BIP-based n-block inequalities restrict (x, z) to the boxes (north-east hatching). The feasible set of the strengthened PWL model is their intersection.

2.4 Tightening the Piecewise-Linear Approximation

For a fixed discretization, the PWL approximation of a nonlinear constraint is a mixed-integer model: continuous variables are linked to binary interval selectors by linear constraints. The projection of this MIP onto the original variables yields the PWL relaxation R^{PWL}. In the simple univariate example of Figure 4, a standard PWL approximation of a function $z = f(x)$ on $[-1, 1]$ with two intervals for x produces two oblique parallelograms in the (x, z)-plane (north-west hatching); these parallelograms represent precisely R^{PWL} for this example.

When this PWL model is embedded into a larger reformulation based on an expression tree, the interval selectors for different occurrences of the same variable or for algebraically related subexpressions are not independent. Each such subexpression induces a BIP structure on the corresponding binary variables, and the associated n-block inequalities express consistency of the interval choices across the tree. Adding these inequalities excludes certain 0–1 patterns that were allowed by the standard PWL model.

In the example of Figure 4, the BIP constrains the set of feasible points to the axis-aligned boxes (north-east hatching). The feasible set of the strengthened PWL relaxation is then the intersection

$$R^{\text{PWL}}_{\text{BIP}} \;=\; R^{\text{PWL}} \cap R^{\square},$$

where $R^{\square}$ denotes the union of these boxes. Only points lying in both the parallelograms and the boxes remain feasible. In this particular example, the region removed by the n-block inequalities is exactly the part of the parallelograms lying below $z = 0$; in general, it is the portion of R^{PWL} that falls outside $R^{\square}$. Since the n-block inequalities are linear constraints on the binaries, they simultaneously tighten the MIP formulation and its linear relaxation.

3 Extensions to the Bipartite Implication Polytope

In the previous section, we observed the appearance of BIP structures when we have subtrees in the expression tree where the root and all leaves correspond to discretized variables. If there are more than two leaves, the implication structure is no longer bipartite but multipartite. In the previous section, the relation mapping Φ was set-valued. This demands an extension to the BIP theory.

3.1 Extension to Multiple Implying Sets

We formally define the problem. At its core is the concept of conditional relations between binary variables across multiple sets, which we represent and analyze through the multipartite implication polytope.

We consider an instance involving $\gamma \in \mathbb{N}$ implying sets, where, without loss of generality, each implying set has cardinality $\alpha \in \mathbb{N}$. From each implying set, a single element must be chosen. For the i-th implying set, the choice of the element is denoted by an index $j_i \in [\alpha]$.

The choices across all implying sets are represented by an index vector $\mathbf{j} = (j_1, \ldots, j_\gamma) \in [\alpha]^\gamma$. The choices in the implying sets imply a corresponding element $l \in [\beta]$ from the implied set. The implication relationship between a potential choice in each implying set and the result in the implied set is defined by a mapping

$$\varphi : [\alpha]^\gamma \to [\beta],\ \mathbf{j} = (j_1, \ldots, j_\gamma) \mapsto l,$$

where $\varphi(\mathbf{j}) = l$ determines the index l of the corresponding element in the implied set. W.l.o.g., we assume the mapping φ is surjective, ensuring every element in the implied set is conditioned by at least one combination of implying set choices. Therefore, the implied set cardinality β is bounded by α^γ. We speak of an multipartite implication instance, if parameters α, γ, β and mapping φ are given.

To analyze the relations mathematically, we model the possible choices with binary vectors. The selection in the i-th implying set is encoded using a binary vector $\mathbf{x}^i \in \{0,1\}^\alpha$, where

$$x^i_j = \begin{cases} 1 & \text{if } j = j_i, \\ 0 & \text{otherwise,} \end{cases} \qquad \text{for all } j \in [\alpha].$$

The selection in the according implied set is represented analogously by a binary vector $\mathbf{y} \in \{0,1\}^\beta$. The requirement to select exactly one element from the i-th implying set is modeled as a multiple choice constraint

$$\sum_{j \in [\alpha]} x^i_j = 1.$$

The binary points fulfilling the multiple choice constraints for all implying sets and the implied set are captured by the set

$$\mathrm{MC} := \left\{ (\mathbf{x}^1, \ldots, \mathbf{x}^\gamma, \mathbf{y}) \in \{0,1\}^{\alpha\gamma+\beta} \;\middle|\; \sum_{j \in [\alpha]} x^i_j = 1 \ \text{ for all } \ i \in [\gamma],\ \sum_{l \in [\beta]} y_l = 1 \right\}.$$

This set ensures that each $\mathbf{x}^i$ represents a valid selection from its corresponding implying set and that $\mathbf{y}$ represents a single element in the implied set. Using the mapping φ, we restrict the set MC to ensure only feasible relations. This is formalized in the set

$$S(\varphi) := \left\{ (\mathbf{x}^1, \ldots, \mathbf{x}^\gamma, \mathbf{y}) \in \mathrm{MC} \;\middle|\; \prod_{i \in [\gamma]} x^i_{j_i} \le y_{\varphi(\mathbf{j})} \ \text{ for all } \ \mathbf{j} = (j_1, \ldots, j_\gamma) \in [\alpha]^\gamma \right\}. \tag{4}$$

The multipartite implication polytope is defined as its convex hull, i.e., $P(\varphi) := \mathrm{conv}(S(\varphi))$. For the sake of clear presentation, we define column vector $\mathbf{z} := ((\mathbf{x}^1)^\top, \ldots, (\mathbf{x}^\gamma)^\top, \mathbf{y}^\top)^\top$. Given an objective vector $\mathbf{w} \in \mathbb{R}^{\alpha\gamma+\beta}$ we state the minimization problem over the set $S(\varphi)$

$$\begin{aligned} \min_{\mathbf{x}^1, \ldots, \mathbf{x}^\gamma, \mathbf{y}} \quad & \mathbf{w}^\top \mathbf{z} \\ \text{s.t.} \quad & (\mathbf{x}^1, \ldots, \mathbf{x}^\gamma, \mathbf{y}) \in S(\varphi). \end{aligned} \tag{5}$$

This is a binary nonlinear optimization problem, falling under the class of MINLP. Such problems are computationally challenging in general, as they involve both nonlinearity and integrality.

In theory, for our case, one could solve (5) by enumerating all vectors in $S(\varphi)$ and picking a vector that results in the smallest objective value. Though, we focus on multipartite implication instances which are substructures of large MINLP problems, where enumeration is not possible.

To obtain a computationally facilitated problem, we want to linearize the nonlinear constraints in problem (5). For that, we employ the McCormick envelope, a convex relaxation technique for the optimization of non-convex NLPs. This relaxation provides a linear approximation for the nonlinear constraints, allowing to redefine the set (4) as

$$S_{\text{lin}}(\varphi) = \left\{ (\mathbf{x}^1, \ldots, \mathbf{x}^\gamma, \mathbf{y}) \in \text{MC} \;\middle|\; \sum_{i \in [\gamma]} x^i_{j_i} \leq y_{\varphi(\mathbf{j})} + \gamma - 1 \text{ for all } \mathbf{j} \in [\alpha]^\gamma \right\}.$$

Note that these sets are the same, therefore,

$$P(\varphi) = \text{conv}(S(\varphi)) = \text{conv}(S_{\text{lin}}(\varphi)).$$

The reformulated optimization problem

$$\begin{aligned} \min_{\mathbf{x}^1, \ldots, \mathbf{x}^\gamma, \mathbf{y}} \quad & \mathbf{w}^\top \mathbf{z} \\ \text{s.t.} \quad & (\mathbf{x}^1, \ldots, \mathbf{x}^\gamma, \mathbf{y}) \in S_{\text{lin}}(\varphi). \end{aligned}$$

is now a binary **linear** problem. However, while both binary sets $S(\varphi)$ and $S_{\text{lin}}(\varphi)$ are equal, their continuous relaxation $\text{cont}(\cdot)$ differ in general. This refers to the relaxation of all binary variables into continuous variables within $[0, 1]$. It holds that $\text{cont}(S(\varphi)) \subseteq \text{cont}(S_{\text{lin}}(\varphi))$.

Now, we explore properties and theoretical results related to the multipartite implication polytope with γ implying sets. Let γ, α, β and φ be arbitrary but fixed. An inequality in the variable space has the form

$$\sum_{i \in [\gamma]} \sum_{j \in [\alpha]} a^i_j x^i_j \leq \sum_{l \in [\beta]} b_l y_l + c, \tag{6}$$

with variables $\mathbf{x}^i \in \mathbb{R}^\alpha$ for all $i \in [\gamma]$ and $\mathbf{y} \in \mathbb{R}^\beta$. The coefficients are $\mathbf{a}^i \in \mathbb{R}^\alpha$ for all $i \in [\gamma]$, $\mathbf{b} \in \mathbb{R}^\beta$, and $c \in \mathbb{R}$.

There is a one-to-one correspondence between all index combinations $\mathbf{j} \in [\alpha]^\gamma$ and all vertices of $P(\varphi)$. This is due to the combinatorial structure of $S(\varphi)$, as any binary point in $S(\varphi)$ gives a vertex of its convex hull $P(\varphi)$. Since every binary point in $S(\varphi)$ can be written as $(\mathbf{x}^1, \ldots, \mathbf{x}^\gamma, \mathbf{y})$ for a $\mathbf{j} \in [\alpha]^\gamma$, we can define each vertex $\mathbf{v}_\mathbf{j} = (\mathbf{x}^1, \ldots, \mathbf{x}^\gamma, \mathbf{y})$. We can also give an explicit representation of the vertices, independent of vectors $\mathbf{x}^i$. The vertex corresponding to a particular $\mathbf{j} \in [\alpha]^\gamma$ is given by

$$\mathbf{v}_\mathbf{j} = \sum_{i \in [\gamma]} \mathbf{e}_{(i-1)\alpha + j_i} + \mathbf{e}_{\alpha\gamma + \varphi(\mathbf{j})} \in \{0, 1\}^{\alpha\gamma + \beta}.$$

The vector $\mathbf{e}_s$ for $s \in [\alpha\gamma + \beta]$ denotes the s-th standard unit vector in $\mathbb{R}^{\alpha\gamma+\beta}$. Thus, we can rewrite set (4) to

$$S(\varphi) = \bigcup_{\mathbf{j} \in [\alpha]^\gamma} \{\mathbf{v}_\mathbf{j}\}.$$

Turning from the inner description by vertices to the outer one by inequalities, we note that an arbitrary inequality is valid for the multipartite implication polytope, if and only if it is valid for every vertex of $P(\varphi)$. We present a useful result to verify the validity of an inequality in the variable space.

Lemma 1. *If an inequality*

$$\sum_{i \in [\gamma]} \sum_{j \in [\alpha]} a^i_j x^i_j \leq \sum_{l \in [\beta]} b_l y_l + c, \tag{7}$$

satisfies

$$b_l \geq \max_{\mathbf{j}\in[\alpha]^\gamma, \varphi(\mathbf{j})=l} \left\{ \sum_{i\in[\gamma]} a^i_{j_i} - c \right\}, \qquad \textit{for all } l \in [\beta], \tag{8}$$

it is valid for $P(\varphi)$*.*

Proof. Let $\mathbf{v_j}$ be a vertex of $P(\varphi)$ corresponding to a particular $\mathbf{j} \in [\alpha]^\gamma$. This vertex is representable as $\mathbf{v_j} = (\mathbf{x}^1, \dots, \mathbf{x}^\gamma, \mathbf{y})$. Inserting these binary vectors into inequality (7) we obtain

$$\sum_{i\in[\gamma]} a^i_{j_i} \leq b_{\varphi(\mathbf{j})} + c.$$

By condition (8), this inequality holds. As vertex $\mathbf{v_j}$ was chosen arbitrarily, (7) is fulfilled for every vertex of $P(\varphi)$, implying it is valid for $P(\varphi)$. □

Building on the characterization of valid inequalities from Lemma 1, we now formalize a definition for a specific set of valid inequalities.

Definition 1. *Let*

$$\sum_{i\in[\gamma]} \sum_{j\in[\alpha]} a^i_j x^i_j \leq \sum_{l\in[\beta]} b_l y_l + c$$

be an inequality in the variable space. We define $\mathcal{S}$ *as the set that includes all such inequalities satisfying the two conditions*

$$b_l = \max_{\mathbf{j}\in[\alpha]^\gamma, \varphi(\mathbf{j})=l} \left\{ \sum_{i\in[\gamma]} a^i_{j_i} - c \right\} \textit{ for all } l \in [\beta], \quad \textit{and} \tag{a}$$

$$a^i_{\max} - a^i_{\min} \leq b_{\max} - b_{\min} \textit{ for all } i \in [\gamma], \tag{b}$$

where $a^i_{\max} := \max_{j\in[\alpha]} a^i_j$ *and* $a^i_{\min} := \min_{j\in[\alpha]} a^i_j$*.*

With condition (a) we restrict $\mathcal{S}$ to inequalities that fulfill the validity characterization (8) with equality. This is a proper restriction, since one can construct a valid inequality for $P(\varphi)$ that fulfills the validity characterization only with strict inequality. For instance, we can increase the value of each coefficient b_l on the right-hand side and maintain validity.

With (b) we enforce another proper condition to the inequalities in $\mathcal{S}$ as we show in the next example.

Example 3. *Set* $\gamma = \alpha = \beta = 2$ *and define the surjective mapping*

$$\varphi : [2]^2 \to [2],\ \mathbf{j} \mapsto \begin{cases} 1 & \textit{if } \mathbf{j} \in \{(1,1),(2,1)\} \\ 2 & \textit{if } \mathbf{j} \in \{(1,2),(2,2)\} \end{cases}.$$

Inequality

$$x^1_2 \leq y_1 + y_2$$

satisfies property (a) *as*

$$b_1 = \max_{\mathbf{j}\in\{(1,1),(2,1)\}} \left\{ a^1_{j_1} + a^2_{j_2} \right\} = a^1_2 + a^2_1 = 1 + 0 = 1,$$

$$b_2 = \max_{\mathbf{j}\in\{(1,2),(2,2)\}} \left\{ a^1_{j_1} + a^2_{j_2} \right\} = a^1_2 + a^2_2 = 1 + 0 = 1,$$

and is, therefore, valid for $P(\varphi)$*. However, property* (b) *does not hold for* $i = 1$*, since*

$$a^1_{\max} - a^1_{\min} = 1 - 0 = 1 > 0 = 1 - 1 = b_{\max} - b_{\min}.$$

Note that, inequalities in $\mathcal{S}$ don't have a unique representation. Besides scaling the inequality's coefficients, we can add or subtract multiple choice constraints

$$
\begin{aligned}
s \cdot \sum_{j\in[\alpha]} x_j^i &= s, \qquad i \in [\gamma], \\
s \cdot \sum_{l\in[\beta]} y_l &= s,
\end{aligned}
$$

for some $s \in \mathbb{R}$ to obtain equivalent inequalities. Another useful property, which we will require later in a proof, is shown in the following lemma.

Lemma 2. *If inequality*

$$\sum_{i\in[\gamma]} \sum_{j\in[\alpha]} a_j^i x_j^i \leq \sum_{l\in[\beta]} b_l y_l + c, \tag{9}$$

satisfies

$$b_l = \max_{\mathbf{j}\in[\alpha]^\gamma, \varphi(\mathbf{j})=l} \left\{ \sum_{i\in[\gamma]} a_{j_i}^i - c \right\}, \qquad \textit{for all } l \in [\beta], \tag{10}$$

it holds true that

$$\sum_{i\in[\gamma]} a_{\max}^i = b_{\max} + c.$$

Proof. "$\leq$": There exists a $\mathbf{j}' \in [\alpha]^\gamma$ such that $a_{j_i'}^i = a_{\max}^i$ for each $i \in [\gamma]$ and $\varphi(\mathbf{j}') = l'$. Inserting the corresponding vertex $\mathbf{v}_{\mathbf{j}'}$ into inequality (9) we obtain

$$\sum_{i\in[\gamma]} a_{\max}^i = \sum_{i\in[\gamma]} a_{j_i'}^i \leq b_{l'} + c \leq b_{\max} + c.$$

"$\geq$": Because φ is surjective, there exists a $\mathbf{j}' \in [\alpha]^\gamma$ such that $\varphi(\mathbf{j}') = l'$ and $b_{l'} = b_{\max}$. Because of condition (10), without loss of generality, it is $b_{l'} = \sum_{i\in[\gamma]} a_{j_i'}^i - c$. Then, it holds that

$$\sum_{i\in[\gamma]} a_{\max}^i \geq \sum_{i\in[\gamma]} a_{j_i'}^i = b_{l'} + c = b_{\max} + c.$$

□

With this prework we can state the main theorem in this section, that allows us to characterize facets of $P(\varphi)$.

Theorem 2. *Let F be a facet of $P(\varphi)$. Then, either F is induced by a lower bound or by an inequality in $\mathcal{S}$.*

Proof. We refer to Appendix A □

Theorem 2 shows that, when considering an inequality in the variable space, which is not a lower bound, properties (a) and (b) from Definition 1 are necessary conditions for being a facet. That is, for an arbitrary mapping φ, all facets of the multipartite implication polytope $P(\varphi)$, which are not induced by lower bounds, are induced by inequalities in $\mathcal{S}$. However, conditions (a) and (b) are not sufficient, since we find inequalities in $\mathcal{S}$ that are not facets. We underline this in the next example.

Example 4. *Suppose we have the setting $\gamma = \beta = 4$, $\alpha = 2$, and mapping*

$$\varphi : [2]^4 \to [4],\ \mathbf{j} \mapsto ((j_1 + j_2 + j_3 + j_4 - 4) \mod 4) + 1.$$

We want to illustrate Definition 1 and Theorem 2. Inequality

$$2x_2^1 + 2x_1^3 \leq -y_1 + y_2 + y_3 + y_4 + 3 \tag{11}$$

fulfills both properties (a) *and* (b) *and belongs, therefore, to $\mathcal{S}$. We find six vertices of $P(\varphi)$, for which (11) holds tight. All six vertices are affine independent, which means that the dimension of the corresponding facet is five. Since facets of $P(\varphi)$ have dimension six, (11) is not a facet defining inequality.*

3.2 Extension to Set-Elements in the Implied Set

We introduce an extension to the multipartite implication polytope where the selection of a combination of elements in the implying sets necessitates the selection of an element from a specific *subset* of the implied set, rather than a single unique element.

Let $\Phi : [\alpha]^\gamma \to 2^{[\beta]} \setminus \{\emptyset\}$ be a set-valued mapping. For a choice of indices $\mathbf{j} \in [\alpha]^\gamma$ in the implying sets, the selected element in the implied set must be contained in the set $\Phi(\mathbf{j})$. The multipartite set implication polytope is defined as the convex hull of valid binary points

$$P^S(\Phi) := \operatorname{conv}\left(\left\{(\mathbf{x}^1, \ldots, \mathbf{x}^\gamma, \mathbf{y}) \in \mathrm{MC} \;\middle|\; \prod_{i\in[\gamma]} x^i_{j_i} \leq \sum_{l\in\Phi(\mathbf{j})} y_l \quad \text{for all } \mathbf{j} \in [\alpha]^\gamma\right\}\right).$$

Here, the product constraint ensures that if the combination $\mathbf{j}$ is active (all $x^i_{j_i} = 1$), then at least one y_l with $l \in \Phi(\mathbf{j})$ must be set to 1. Since $\mathbf{y}$ is constrained by MC to have exactly one non-zero entry, this enforces the set implication logic.

We now generalize the concept of valid inequalities for this structure. In the standard case, we examined inequalities based on variable coefficients $\mathbf{a}^i$ for the implying sets. Here, we adapt this to define multipartite set-valid inequalities. Consider an inequality of the form (6). To ensure validity and tightness, the coefficients b_l for the implied set must be sufficiently large to cover the *pressure* exerted by the implying coefficients $\mathbf{a}^i$. Analogous to the construction in the standard multipartite case, we define the coefficients for the implied set variable y_l based on the maximum contribution from any implying configuration $\mathbf{j}$ that allows l.

Definition 2. *We call an inequality of the form* (6) *a general MPIP inequality if it satisfies the two conditions*

$$b_l = \max_{\mathbf{j}\in[\alpha]^\gamma, l\in\Phi(\mathbf{j})} \left\{\sum_{i\in[\gamma]} a^i_{j_i} - c\right\} \text{ for all } l \in [\beta], \quad \text{and} \tag{aSet}$$

$$a^i_{\max} - a^i_{\min} \leq b_{\max} - b_{\min} \text{ for all } i \in [\gamma]. \tag{bSet}$$

Lemma 3. *All general MPIP inequalities are valid for $P(\Phi)$.*

Proof. The vertices of $P^S(\Phi)$ are given by the set

$$V(\Phi) = \left\{\mathbf{v}_{\mathbf{j},l} \in \{0,1\}^{\alpha\gamma+\beta} \mid \mathbf{j} \in [\alpha]^\gamma,\ l \in \Phi(\mathbf{j})\right\},$$

where

$$\mathbf{v}_{\mathbf{j},l} = \sum_{i\in[\gamma]} \mathbf{e}_{(i-1)\alpha+j_i} + \mathbf{e}_{\alpha\gamma+l} \in \{0,1\}^{\alpha\gamma+\beta}$$

is the vertex corresponding to selecting indices $\mathbf{j}$ and implied element l. Consider an arbitrary vertex $\mathbf{v}_{\mathbf{j},l} \in V(\Phi)$. The left-hand side of the inequality (6) evaluates to $\sum_{i\in[\gamma]} a^i_{j_i}$. The right-hand side evaluates to $b_l + c$. Since $l \in \Phi(\mathbf{j})$, by definition (aSet), we have

$$b_l \geq \sum_{i\in[\gamma]} a^i_{j_i} - c.$$

Rearranging yields $\sum_{i\in[\gamma]} a^i_{j_i} \leq b_l + c$, which proves validity. $\square$

We now characterize the facet-defining inequalities of the set implication polytope.

Theorem 3. *Let F be a facet of $P^S(\Phi)$. Then, either F is induced by a lower bound or by an inequality satisfying* (aSet) *and* (bSet)*.*

Proof. The proof proceeds analogously to that of Theorem 2. $\square$

Separation for the multipartite set implication polytope can be performed similarly to the bipartite case. Given a fractional point $(\bar{\mathbf{x}}^1, \dots, \bar{\mathbf{x}}^\gamma, \bar{\mathbf{y}})$, the separation problem seeks coefficients of a general MPIP inequality that is violated by the point:

$$\begin{aligned} \max_{\mathbf{a}^1,\dots,\mathbf{a}^\gamma,\mathbf{b}} \quad & \sum_{i\in[\gamma]} (\mathbf{a}^i)^\top \bar{\mathbf{x}}^i - \sum_{l\in[\beta]} b_l \bar{y}_l - (\gamma - 1) \\ \text{s.t.} \quad & b_l \geq \sum_{i\in[\gamma]} a^i_{j_i} - (\gamma - 1) \quad \forall\, \mathbf{j} \in [\alpha]^\gamma, \forall\, l \in \Phi(\mathbf{j}), \\ & \mathbf{a}^i \in [0,1]^\alpha \quad \forall\, i \in [\gamma], \\ & \mathbf{b} \in [0,1]^\beta \end{aligned} \tag{12}$$

The following theorem shows that optimal vertices of this LP yield valid inequalities satisfying both defining conditions. Note that the theorem requires all entries of the point to be strictly positive. For a point with zero entries, this assumption can be satisfied by slightly perturbing the entries to obtain a nearby strictly positive point.

Theorem 4. *Let $(\bar{\boldsymbol{x}}^1, \dots, \bar{\boldsymbol{x}}^\gamma, \bar{\boldsymbol{y}}) > 0$ be a strictly positive point that fulfills $\sum_{j\in[\alpha]} \bar{x}^i_j = 1$ for all $i \in [\gamma]$ and $\sum_{l\in[\beta]} \bar{y}_l = 1$. Then every vertex $(\tilde{\mathbf{a}}^1, \dots, \tilde{\mathbf{a}}^\gamma, \tilde{\mathbf{b}})$ of the set of feasible points that maximizes the separation LP* (12) *yields coefficients satisfying conditions* (aSet) *and* (bSet)*.*

Proof. Let $(\tilde{\mathbf{a}}^1, \dots, \tilde{\mathbf{a}}^\gamma, \tilde{\mathbf{b}})$ be an optimal solution of (12). The constraints of the separation LP imply

$$\tilde{b}_l \geq \max_{\mathbf{j}\in[\alpha]^\gamma : l\in\Phi(\mathbf{j})} \left\{ \sum_{i\in[\gamma]} \tilde{a}^i_{j_i} - (\gamma - 1) \right\} \quad \forall\, l \in [\beta].$$

Since $\bar{y}_l > 0$ for all $l \in [\beta]$, any increase of $\tilde{b}_l$ beyond this maximum would decrease the objective value (as we maximize $-\sum_l b_l \bar{y}_l$). Therefore, at optimality, the inequality must hold with equality. This establishes condition (aSet).

To prove condition (bSet), we first show $\tilde{a}^i_{\max} = \tilde{b}_{\max} = 1$ for all $i \in [\gamma]$. We know $\sum_{i\in[\gamma]} \tilde{a}^i_{\max} - (\gamma - 1) = \tilde{b}_{\max}$, otherwise we could decrease $\tilde{b}_{\max}$ while staying feasible and increasing the objective value. Assume $\tilde{a}^{i'}_{\max} = 1 - \epsilon$ for an arbitrary $i' \in [\gamma]$, $0 < \epsilon < 1$. Then we can use the multiple-choice condition on the point $(\bar{\mathbf{x}}^1, \dots, \bar{\mathbf{x}}^\gamma, \bar{\mathbf{y}})$ to both add and subtract ϵ from all entries of $\tilde{\mathbf{a}}^{i'}$ and $\tilde{\mathbf{b}}$, while staying feasible and without altering the objective value. This contradicts the assumption that $(\tilde{\mathbf{a}}^1, \dots, \tilde{\mathbf{a}}^\gamma, \tilde{\mathbf{b}})$ was a vertex. $\square$

Example 5 (Example 1 continued). *We return to the piecewise constant relaxation of $\xi = \chi^1\chi^2$ with the relation mapping Φ established above in Example 1. The pure binary feasible set associated with the piecewise constant relaxation is*

$$S^{\text{SET}}(\Phi) := \left\{(x^1, x^2, y) \in M \;\middle|\; x^1_{j_1} x^2_{j_2} \leq \sum_{l \in \Phi(j_1,j_2)} y_l,\ \forall (j_1, j_2) \in [4]^2\right\},$$
$$M := \left\{(x^1, x^2, y) \in \{0,1\}^{12} \mid \sum_{j_1=1}^{4} x^1_{j_1} = \sum_{j_2=1}^{4} x^2_{j_2} = \sum_{l=1}^{4} y_l = 1\right\}.$$

This set coincides with the integer points of an extended MPIP. One general MPIP inequality is

$$x^1_3 + 2x^1_4 + x^2_3 + 2x^2_4 \leq y_2 + 2y_3 + 2y_4 + 2.$$

It cuts off the non-integer point

$$s_{x^1} = (0, \tfrac{1}{3}, \tfrac{1}{3}, \tfrac{1}{3}), \quad s_{x^2} = (0, 0, \tfrac{1}{3}, \tfrac{2}{3}), \quad s_y = (1, 0, 0, 0).$$

Indeed, evaluating the inequality at this point yields

$$LHS = \tfrac{1}{3} + 2 \cdot \tfrac{1}{3} + \tfrac{1}{3} + 2 \cdot \tfrac{2}{3} = \tfrac{8}{3}, \qquad RHS = 0 + 0 + 0 + 2 = 2,$$

so the inequality is violated. At the same time, its validity can be verified by checking all 24 *integer points in $S^{\text{SET}}(\Phi)$. This illustrates how the BIP-based n-block inequalities strengthen the relaxation obtained from the naive discretization of $\xi = \chi^1\chi^2$.*

4 General MPIP Inequalities for Piecewise-Linear Relaxations of General Nonlinear Expressions

In this chapter, we describe how the MPIP theory can be applied to piecewise-linear relaxations. As discussed in Section 2.2, PWL-based MIP formulations introduce auxiliary binary variables to identify the active segment of each discretized variable.

In such formulations, the selection of a segment for one variable restricts the feasible segments of other variables whenever they jointly appear in a nonlinear expression. Consequently, the choice of binary variables for one or more discretized variables can limit the admissible choices for others. This dependency structure is a key prerequisite for the occurrence of MPIP patterns.

In particular, we consider the case where a discretized variable is restricted through two other discretized variables. This can happen when all these discretized variables occur in the same nonlinear expression. Assuming an expression tree for this nonlinearity, if one variable ξ is a parent node to both, χ^1 and χ^2, then we say that the choice of χ^1 and χ^2 restricts the possible choices for ξ and this naturally forms an MPIP structure.

A motivating example is given in the following.

Example 6. *Let us assume that the nonlinear term*

$$\sin\left(z_1^2 + z_2^3\right) = z_3$$

is contained in some MINLP. When creating a piecewise-linear relaxation, this term is first reformulated using expression trees, resulting in

$$\begin{aligned} \sin(z_4) &= z_3, \\ z_4 &= z_5 + z_6, \\ z_5 &= z_1^2, \\ z_6 &= z_2^3. \end{aligned}$$

Then, the variables contained in nonlinear equations are discretized. Here, this is z_1, z_2 *and* z_4*. We assume bounds* $-4 \leq z_1 \leq 4$ *and* $-2 \leq z_2 \leq 2$*. Then, also* z_4 *is bounded by* $-8 \leq z_4 \leq 24$*.*

As one can see, the variables z_1 *and* z_2 *are child nodes of* z_4 *and, thus, we have an MPIP structure here when using*

$$\chi^1 := z_1,\ \chi^2 := z_2,\ \xi := z_4.$$

For the discretization, we define equidistant breakpoints for χ^1*,* χ^2 *and* ξ*:*

$$\begin{aligned} \eta_0^1 = -4,\ \eta_1^1 = -3,\ \dots,\ \eta_8^1 = 4, \\ \eta_0^2 = -2,\ \eta_1^2 = -1.5,\ \dots,\ \eta_8^2 = 2, \\ \tau_0 = -8,\ \tau_1 = -4,\ \dots,\ \tau_8 = 24. \end{aligned}$$

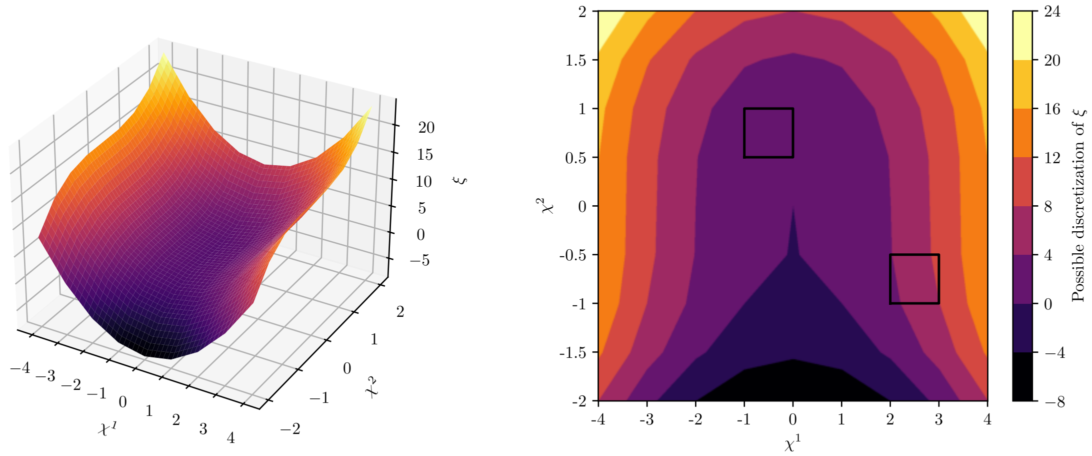
For identification of the active segment, we use the one-hot encoded vectors

$$\boldsymbol{x}^1 \in \{0,1\}^8,\ \boldsymbol{x}^2 \in \{0,1\}^8 \text{ and } \boldsymbol{y} \in \{0,1\}^8.$$

Further, let $\bar{f}_1(\chi^1)$*,* $\bar{f}_2(\chi^2)$ *and* $\bar{f}_3(\xi)$ *be the piecewise-linear relaxations of* x_1^2*,* x_2^3 *and* $\sin(z)$*. Then,we obtain*

$$\begin{aligned} \bar{f}_3(\xi) &= z_3, \\ \xi &= z_5 + z_6, \\ z_5 &= \bar{f}_1(\chi^1), \\ z_6 &= \bar{f}_2(\chi^2), \end{aligned} \tag{13}$$

where z_3, z_5 *and* z_6 *are continuous non-discretized variables.*



(a) Continuous values for ξ (b) Discrete values for ξ

Figure 5: Visualization for $\xi = \bar{f}_1(\chi^1) + \bar{f}_2(\chi^2)$ where $\bar{f}_1(\chi^1)$ and $\bar{f}_2(\chi^2)$ are the piecewise-linear relaxations for z_1^2 and z_2^3. In 5a, the continuous result is given. Contrary, in 5b, the plot is colored depending on the segment in which ξ lies when it is discretized.

Figure 5a shows a plot for $\xi = \bar{f}_1(\chi^1) + \bar{f}_2(\chi^2)$ *of System 13. Figure 5b shows the same term, but the values of* ξ *are colored depending on the segment where* ξ *lies when evaluating* $\bar{f}_3(\xi)$*. Each segment is defined by some binary variable. So, the choices of the segments for* χ^1 *and* χ^2 *restrict the feasible segment choices for* ξ*. Two possible segment choices are shown in Figure 5b:*

- *Segment choice* $x_3^1 = 1$*, i.e.,* $-1 \leq \chi^1 \leq 0$ *and* $x_5^2 = 1$*, i.e.,* $0.5 \leq \chi^2 \leq 1$*. This results in* $0.125 \leq \xi \leq 2$*, and, thus,* ξ *can only lie in the segment between* $\tau_2 = 0$ *and* $\tau_3 = 4$*.*
- *Segment choice* $x_6^1 = 1$*, i.e.,* $2 \leq \chi^1 \leq 3$ *and* $x_2^2 = 1$*, i.e.,* $-1 \leq \chi^2 \leq -0.5$*. This results in* $3 \leq \xi \leq 8.875$*, and, thus,* ξ *can lie in all segments between* $\tau_2 = 0$ *and* $\tau_5 = 12$*.*

In terms of the MPIP, this means $\Phi(3,5) = \{2\}$ *and* $\Phi(6,2) = \{2,3,4\}$*. Further values of* Φ *can be determined equivalently.*

In the following, we introduce different MIP reformulations. These reformulations differ – besides others – in how the segments are modeled using binary variables.

4.1 Multiple-Choice Model

The multiple-choice model [7] uses a quite simple idea: A vector $\mathbf{x}$ containing n binary variables indicates the segment in which a point lies. Further, n continuous variables $\hat{\chi}_1, \ldots, \hat{\chi}_n$ are used to model the exact value of χ. The sum of all $x_i \cdot \hat{\chi}_i$, i.e., only the value of the variable $\hat{\chi}_i$ where the corresponding binary variable x_i is active, then equals χ.

Recall that the binary variables are defined by

$$\begin{aligned} \sum_{i=1}^{n} x_i &= 1, && && \text{(3c rev.)} \\ x_i \eta_{i-1} &\leq \hat{\chi}_i && \text{for all } i \in [n], && \text{(3d rev.)} \\ \hat{\chi}_i &\leq x_i \eta_i && \text{for all } i \in [n], && \text{(3e rev.)} \end{aligned}$$

and, thus, are one-hot encoded and can directly be used for modeling an MPIP structure.

4.2 Classical Incremental Model

The classical incremental method also uses a vector $\hat{\mathbf{x}} \in \{0,1\}^n$ of n binary variables. In contrast to the multiple-choice method, this vector is not one-hot encoded. Instead, if segment i is active, the variable $\hat{x}_i$ and all variables $\hat{x}_{i^-}$ with $i^- < i$ are active, while all variables $\hat{x}_{i^+}$ with $i_+ > i$ are inactive. For all continuous variables δ it holds that $\delta_{i^-} = 1$ and $\delta_{i^+} = 0$. The variable δ_i can take any value between $0 \leq \delta_i \leq 1$ and indicates how far the point lies in the interval. Recall the definition of the binary variables in the classical incremental method:

$$\begin{aligned} \delta_1 &\leq 1, && && \text{(2c rev.)} \\ \delta_{i+1} &\leq \hat{x}_i && \text{for all } i \in [n-1], && \text{(2d rev.)} \\ \hat{x}_i &\leq \delta_i && \text{for all } i \in [n], && \text{(2e rev.)} \\ \delta_n &\geq 0. && && \text{(2f rev.)} \end{aligned}$$

As $\hat{\mathbf{x}}$ is not one-hot encoded we cannot apply the MPIP theory directly. Instead, a new binary vector $\mathbf{x}$ can be created:

$$\begin{aligned} x_1 &\coloneqq 1 - \hat{x}_1, \\ x_i &\coloneqq \hat{x}_{i-1} - \hat{x}_i && \text{for all } i \in \{2, ..., n-1\}, \\ x_n &\coloneqq \hat{x}_{n-1}. \end{aligned}$$

Now, these variables again indicate the active segment directly and can be used for MPIP theory.

5 Precomputing General MPIP Inequalities for Specific Applications

While the cutting plane algorithm described in previous sections is designed to identify and separate violated inequalities dynamically during the branch-and-bound process, specific application domains often exhibit predictable, repetitive structures in their relation matrices. In such cases, relying solely on a generic separation oracle is computationally inefficient. Instead, we can exploit the known analytical properties of the underlying functions to precompute families of strong valid inequalities—specifically corner inequalities and stripe inequalities—and add them to the model formulation *a priori*. This approach tightens the formulation at the root node and reduces the computational overhead associated with the separation routine. We demonstrate this methodology using two classes of problems frequently encountered in the MINLPLib: pooling problems and kriging peaks functions.

$i = 2$

Φ ($i=1$ \ $i=2$)	1	2	3	4	5	6
1	{1}	{1}	{1}	{1}	{1}	{1}
2	{1}	{1}	{1}	{1,2}	{1,2}	{1,2}
3	{1}	{1}	{1,2}	{1,2}	{2,3}	{2,3}
4	{1}	{1,2}	{1,2}	{2,3}	{2,3,4}	{3,4}
5	{1}	{1,2}	{2,3}	{2,3,4}	{3,4,5}	{4,5}
6	{1}	{1,2}	{2,3}	{3,4}	{4,5}	{5,6}

$x^1_5 + x^1_6 + x^2_3 + x^2_4 + x^2_5 + x^2_6 \leq \sum_{l \in [6]\setminus\{1\}} y_l + 1$

$x^1_3 + x^1_4 + x^1_5 + x^1_6 + x^2_5 + x^2_6 \leq \sum_{l \in [6]\setminus\{1\}} y_l + 1$

Figure 6: Example Corner Inequalities

$i = 2$

Φ ($i=1$ \ $i=2$)	1	2	3	4	5	6
1	{4,5,6}	{3,4}	{2,3}	{1,2}	{1}	{1}
2	{4,5,6}	{3,4}	{2,3}	{1,2}	{1}	{1}
3	{4,5,6}	{3,4}	{1,2,3}	{1,2}	{1}	{1}
4	{4,5,6}	{3,4}	{1,2,3}	{1}	{1}	{1}
5	{4,5,6}	{3,4}	{1,2,3}	{1}	{1}	{1}
6	{4,5,6}	{3,4}	{1,2,3}	{1}	{1}	{1}

$\sum_{j \in [6]} x^1_j + x^2_4 + x^2_5 + x^2_6 \leq y_1 + y_2 + 1$

$\sum_{j \in [6]} x^1_j + x^2_1 \leq y_4 + y_5 + y_6 + 1$

Figure 7: Example Stripe Inequalities

5.1 Pooling Problems

Pooling problems are frequent in chemical applications, modeling the blending of intermediate products in storage tanks. These problems are characterized by bilinear expressions representing the product of flow and proportion, denoted as $\xi = \chi_1 \cdot \chi_2$. Extensive research has been conducted on the global optimization of these problems, notably by [1].

When utilizing piecewise-linear approximations for these bilinear terms, the resulting relation matrix φ exhibits a distinct monotonic or staircase structure. This monotonicity arises because, for positive domains, as the values of operands χ and χ increase, the product ξ generally increases. Consequently, the level sets of the relation mapping form contiguous rectangular regions in the discretized domain. To capture this structure efficiently, we utilize corner inequalities. A corner inequality is derived by identifying a rectangular block of indices in the implying sets—the discretized domains of the operands—and determining the minimal set of implied indices required to satisfy the relation.

We illustrate this concept with the instance *pooling_adhya1pq*, as shown in Figure 6. This instance involves the constraint $y = \chi_1 \cdot \chi_2$ with variable bounds $\chi_1 \in [0, 1]$, $\chi_2 \in [0, 30]$, and $\xi \in [0, 30]$. The grid illustrates the mapping φ, where the valid entries naturally cluster, leaving empty or forbidden regions in the corners. For example, the bottom-right region of the matrix corresponds to high values for both χ_1 and χ_2. We use the binary variables $x^1_{j_1}$ for $j_1 \in [\alpha]$, $x^2_{j_2}$ for $j_2 \in [\alpha]$ and y_l for $l \in [\beta]$. In the specific block illustrated, defined by indices $i \in \{5, 6\}$ for x^1 and $j \in \{3, 4, 5, 6\}$ for x^2, the product strictly maps indices that are not 1. The corresponding corner inequality,

$$x^1_5 + x^1_6 + x^2_3 + x^2_4 + x^2_5 + x^2_6 \leq \sum_{l \in [6]\setminus\{1\}} y_l + 1$$

, enforces that if the binary variables corresponding to this high-input block are active, the output variable ξ cannot take the value in the first interval. Similarly, the top-left region captures the implication that low input values necessitate low output values. By precomputing these corner inequalities, we effectively cut off the rectangular regions of the binary hypercube that do not map to the selected output range, enforcing the monotonic geometry of the bilinear function directly in the constraints.

5.2 Kriging Peaks

A second class of problems suitable for precomputation involves Gaussian process regression, often referred to as kriging. In optimization problems such as the peaks function benchmark described by [17], reduced-space formulations are employed where intermediate variables are substituted out. These instances frequently contain terms involving the Euclidean norm, such as

$$f(\chi_1, \chi_2) = \sqrt{a(\chi_1)^2 + b(\chi_2)^2}.$$

Unlike the monotonic staircase structure observed in pooling problems, the relation matrices for these functions often exhibit a stripe pattern. This pattern emerges when the function value is significantly more sensitive to one variable than the other, or when the discretization grid aligns such that specific columns in the relation matrix map to a small, constant subset of implied values regardless of the row choice. To exploit this structure, we introduce stripe inequalities. These are a generalization of corner inequalities where the block extends across the entire domain of one implying variable.

Consider the example from instance *kriging_peaks-red020* presented in Figure 7, which approximates the term $\xi = 0.73(\chi_1)^2 + 9.84(\chi_2)^2$ with bounds $\chi_1 \in [-0.80, 0.20]$, $\chi_2 \in [-0.83, 0.16]$, and $\xi \in [0, 7.29]$. We again use the binary variables $x^1_{j_1}$ for $j_1 \in [\alpha]$, $x^2_{j_2}$ for $j_2 \in [\alpha]$ and y_l for $l \in [\beta]$. The visualization reveals vertical stripes where the valid set of output indices is determined almost exclusively by x^2. For instance, when x^2 is active for one of the indices in $\{4, 5, 6\}$, the output y is constrained to the lower indices $\{1, 2\}$. This holds true regardless of the active binary x^1. The resulting stripe inequality,

$$\underbrace{\sum_{j_1 \in [6]} x^1_{j_1}}_{=1} + x^2_4 + x^2_5 + x^2_6 \leq y_1 + y_2 + 1,$$

sums over all indices of x^1, effectively treating it as a constant contribution, and restricts x^2 to the specific subset. A second stripe inequality is visible for the first column, where fixing x^2 to index 1 forces the output into the higher range. By detecting these column-wise or row-wise dependencies during the preprocessing phase, we can generate a compact set of inequalities that capture the dominant structural features of the function, significantly aiding the solver in pruning the search space.

6 Computational Study

Implementation Environment All computational experiments were conducted on a virtual machine equipped with an Intel Xeon processor with 28 cores operating at 2.70 GHz base frequency and 64 GB of RAM. We employed Gurobi Optimizer version 13.0.0 as the mixed-integer programming solver, configured with default parameter settings. All model implementations and data processing were developed in Python 3.11.13. For the computational study, we developed a git repository for reading MINLP-Lib instances, decomposing all expressions to one-dimensional forms, and creating a piecewise-linear relaxation with either the multiple-choice or the incremental method. The solver, named alpaca (Adaptive Linear Piecewise Approximation with Combinatorial Augmentation), is public and accessible under `https://github.com/utnopt/alpaca`.

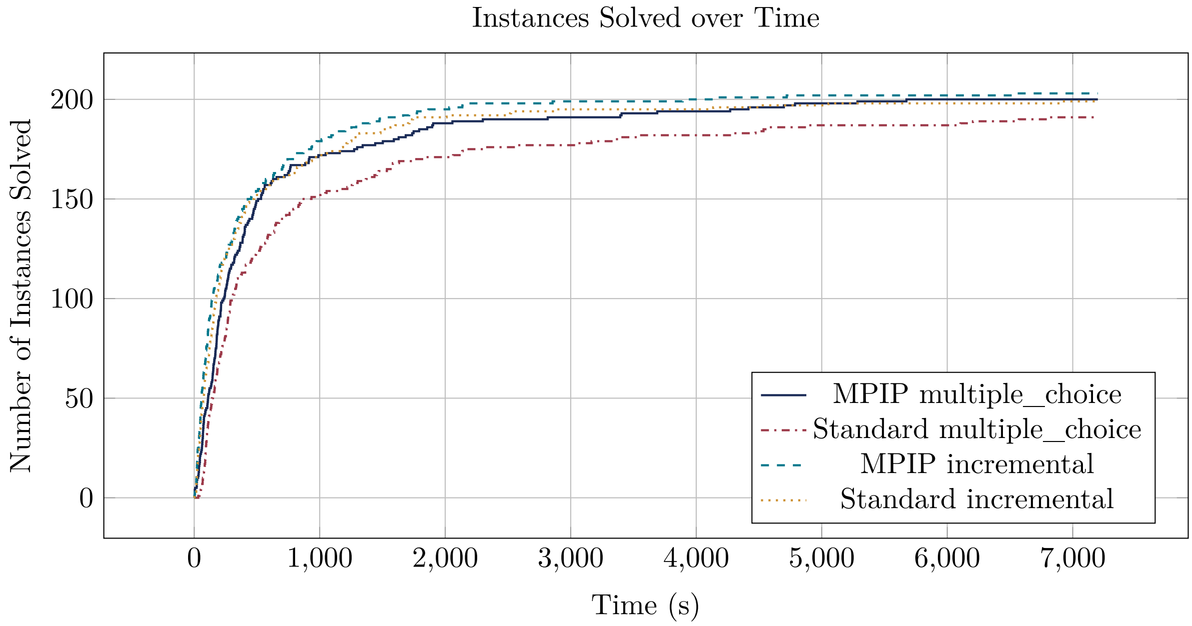


Figure 8: Number of instances solved over time for different methods. Total number of instances: 236

6.1 General MPIP Inequality Separation Routine for MINLP-Lib

We first evaluate the performance of the proposed general MPIP inequality separation routine (12) on a general set of 236 instances from the MINLP-Lib [21]. Instances are selected based on two criteria: the presence of MPIPs and the requirement that all discretized variables are bounded after optimization-based bound tightening. The goal is to assess whether the dynamic separation of General MPIP Inequalities improves the solvability of piecewise-linear MIP reformulations compared to the standard formulations without these valid inequalities. We compare four configurations: the standard multiple-choice formulation, the standard incremental formulation, and their respective counterparts enhanced with the MPIP separation routine.

Figure 8 illustrates the number of solved instances over time for the considered methods. The results indicate a clear advantage for the MPIP-enhanced formulations. The MPIP multiple-choice method, represented by the solid line, solved significantly more instances within the first 1000 seconds compared to the standard multiple-choice approach. Specifically, while the standard multiple-choice method solves 151 instances in this timeframe, the MPIP variant solves 172. This suggests that the separation of General MPIP Inequalities effectively tightens the relaxation early in the branch-and-bound tree, allowing the solver to prune nodes more efficiently.

A similar trend is observed for the incremental formulation, where the MPIP variant (dashed line) consistently lies above the standard incremental formulation (dotted line). Although the standard formulation eventually solves a comparable total number of instances as the time limit of 7200 seconds approaches, we observe superior performance of the MPIP methods in the low-to-medium runtime range. After 1000 seconds the standard formulation solves 172 instances, while the MPIP variant solves 179. The overhead introduced by the separation routine appears to be well-compensated by the reduction in the search space.

Notably, the introduction of the separation routine renders the multiple-choice method competitive with the incremental method. This aligns with the computational study in [3], which observed that while the multiple-choice method effectively generates high-quality primal solutions, it typically falls short to the incremental method in providing dual bounds and closing the optimality gap. The separation of general MPIP inequality appears to bridge this deficiency, significantly enhancing the multiple-choice method by addressing this specific weakness.

6.2 Precomputed General MPIP Inequalities for Specific Application Instances

In addition to the dynamic separation routine, we analyzed the impact of adding precomputed application-specific inequalities—corner inequalities for pooling problems and stripe inequalities for kriging peaks functions—directly to the model formulation.

6.2.1 Corner Inequalities for Pooling

The results for the pooling problem instances are summarized in Table 1. The inclusion of corner inequalities leads to a substantial reduction in computational time across the majority of the test set. We observe an overall geometric mean speedup of a factor of 3.23 when comparing the MPIP method to the standard approach.

The performance improvement is particularly striking for the *digabel* instances. For example, in *pooling_digabel16* and *pooling_digabel19*, the standard approach fails to solve the instances within the 5 hour time limit, whereas the MPIP formulation solves them in under 40 seconds. This corresponds to speedup factors of over 900 and 400, respectively. These results confirm that the corner inequalities effectively capture the monotonic structure of the bilinear terms, preventing the solver from exploring infeasible regions that satisfy the standard relaxation but violate the logical implications of the discretization. While there are a few instances such as *pooling_foulds5tp* where the overhead of the cuts results in a slowdown (speedup < 1), the dramatic gains on the harder instances validate the effectiveness of the precomputation strategy for pooling problems.

6.2.2 Stripe Inequalities for Kriging-Peaks

Table 2 presents the computational results for the kriging peaks function instances. Here, the precomputed stripe inequalities provide a consistent improvement, achieving an overall geometric mean speedup of a factor of 2.58.

The instance *kriging_peaks-red020* demonstrates the potential of this approach, with the runtime decreasing from over $11,000$ seconds to approximately $1,170$ seconds, yielding a speedup of 9.49. Similarly, *kriging_peaks-red050* sees a six-fold improvement in solution time. The stripe inequalities exploit the property that certain variables in the Gaussian process regression term dominate the function value, creating stripes in the relation matrix that can be used to find general MPIP inequalities efficiently. Unlike the pooling set, where some instances exhibited slowdowns, the MPIP method for kriging remains competitive even in cases where the speedup is minimal, such as *kriging_peaks-red500*, where the performance is essentially identical to the standard approach. This suggests that stripe inequalities are a safe and generally beneficial addition to the formulation for this class of problems.

7 Conclusion

In this paper, we have demonstrated that piecewise-linear relaxations of mixed-integer nonlinear programs give rise to rich combinatorial structures that can be systematically exploited to tighten the underlying mixed-integer formulations. Starting from the bipartite implication polytope introduced in earlier work, we identified its occurrence not only in isolated bilinear terms but also throughout expression-tree based reformulations of general nonlinear constraints. This observation motivated the development of the multipartite implication polytope, which accommodates an arbitrary number of implying sets, as well as a natural set-valued extension that captures situations where multiple implied indices are admissible for a given combination of implying choices.

Instance	Standard (s)	MPIP (s)	Speedup
pooling_digabel16 | 20	18000.00	19.60	918.56
pooling_digabel19 | 20	18000.00	37.71	477.30
pooling_digabel18 | 20	13817.49	289.88	47.67
pooling_adhya1stp | 20	1897.33	49.51	38.32
pooling_rt2stp | 20	4956.50	132.65	37.37
pooling_rt2pq | 20	940.12	27.85	33.76
pooling_sppb0tp | 10	14992.52	539.55	27.79
pooling_rt2tp | 20	455.81	25.48	17.89
pooling_sppa0tp | 10	873.28	69.90	12.49
pooling_adhya2stp | 30	1513.61	129.34	11.70
pooling_sppa0stp | 10	3435.23	308.44	11.14
pooling_adhya1pq | 30	1736.99	172.85	10.05
pooling_adhya2pq | 30	216.87	22.16	9.78
pooling_sppb0stp | 10	18000.00	1972.64	9.12
pooling_adhya3stp | 20	242.89	27.75	8.75
pooling_sppa0pq | 10	379.59	55.69	6.82
pooling_adhya4pq | 30	652.19	95.99	6.79
pooling_sppb0pq | 10	605.95	201.00	3.01
pooling_adhya4stp | 20	74.59	29.92	2.49
pooling_sppa9pq | 20	1536.66	824.82	1.86
pooling_foulds4stp | 30	1143.95	622.27	1.84
pooling_adhya1tp | 30	202.67	112.97	1.79
pooling_sppc3tp | 10	3173.43	1927.30	1.65
pooling_foulds4pq | 40	1384.20	879.08	1.57
pooling_sppa9tp | 10	146.64	98.12	1.49
pooling_sppb2pq | 10	477.28	326.18	1.46
pooling_sppa9stp | 10	274.91	194.96	1.41
pooling_sppc0pq | 5	35.92	25.81	1.39
pooling_sppc3pq | 10	2129.82	1580.48	1.35
pooling_sppc1stp | 5	342.99	257.31	1.33
pooling_adhya3tp | 40	376.08	284.29	1.32
pooling_sppc0stp | 5	113.28	90.33	1.25
pooling_sppa5stp | 10	267.75	218.10	1.23
pooling_sppc0tp | 5	37.49	30.84	1.22
pooling_sppc3stp | 5	214.66	177.97	1.21
pooling_sppc1pq | 5	71.72	63.48	1.13
pooling_adhya2tp | 40	95.98	86.39	1.11
pooling_sppc1tp | 5	76.84	73.11	1.05
pooling_sppa5tp | 10	83.36	80.48	1.04
pooling_sppb2tp | 10	335.75	333.33	1.01
pooling_foulds5stp | 30	2096.71	2115.22	0.99
pooling_sppb2stp | 10	892.22	945.59	0.94
pooling_foulds4tp | 30	396.99	433.25	0.92
pooling_sppa5pq | 10	60.58	68.58	0.88
pooling_foulds3tp | 30	418.55	538.62	0.78
pooling_adhya4tp | 30	146.29	191.32	0.76
pooling_foulds3stp | 30	1195.89	1833.84	0.65
pooling_foulds5tp | 30	685.69	1332.53	0.51
pooling_foulds5pq | 40	1182.46	2316.81	0.51
pooling_foulds3pq | 30	129.14	415.78	0.31
Overall	**612.09**	**186.14**	**3.23**

Table 1: Geometric mean runtime (s) comparison between Standard and MPIP methods over all seeds.

Instance	**Standard (s)**	**MPIP (s)**	**Speedup**
kriging_peaks-red020 | 20	11098.92	1169.68	9.49
kriging_peaks-red030 | 20	1393.52	152.33	9.15
kriging_peaks-red050 | 20	7091.18	1143.28	6.20
kriging_peaks-red010 | 20	26.07	22.05	1.18
kriging_peaks-red200 | 10	3251.68	2869.09	1.13
kriging_peaks-red100 | 10	739.11	685.65	1.08
kriging_peaks-red500 | 5	10155.65	10300.34	0.99
Overall	**1833.97**	**710.07**	**2.58**

Table 2: Geometric mean runtime (s) comparison between Standard and MPIP methods over all seeds.

For both the multipartite implication polytope and its set-valued variant, we established a complete characterization of the nontrivial facets. Specifically, we showed that every facet not induced by a variable bound belongs to the class of general MPIP inequalities, which arise from nested block structures in the relation mapping. This characterization provides the theoretical foundation for the design of separation algorithms that operate directly on the binary interval-selection variables introduced by the piecewise-linear reformulation.

On the algorithmic side, we developed a generic separation routine for general MPIP inequalities and demonstrated its integration into standard multiple-choice and incremental piecewise-linear formulations. The separation problem reduces to a linear program whose size depends on the discretization granularity, making it amenable to efficient implementation within a branch-and-cut framework. Furthermore, we exploited structural properties of specific application classes to precompute families of strong valid inequalities a priori. For pooling problems, the monotonic staircase pattern of the relation matrix led to corner inequalities, while for kriging peaks instances the dominance of certain variables resulted in stripe inequalities. Both families can be generated during a preprocessing phase and added to the formulation before the solver is invoked.

The computational study on a broad subset of instances from the MINLPLib confirmed the practical value of the proposed approach. The dynamic separation of general MPIP inequalities yielded consistent improvements across a diverse set of problems, with the enhanced multiple-choice formulation becoming competitive with the incremental method. For the application-specific instances, the precomputed inequalities led to substantial speedups, with geometric mean improvements exceeding a factor of three for pooling problems and a factor of two and a half for kriging peaks functions. Particularly noteworthy is the observation that several instances which could not be solved within a five-hour time limit by the standard formulation were solved in under one minute after adding the proposed cuts.

Several directions for future research emerge from this work. First, the present study focused on the multiple-choice and incremental piecewise-linear reformulations; applying the multipartite implication polytope theory to other formulations, such as convex combination methods remains an open task that may reveal further structural insights and algorithmic opportunities. This might be especially interesting when considering its logarithmic variants. Second, while the facet characterization established here is complete in a theoretical sense, identifying subclasses of general MPIP inequalities that admit closed-form coefficient descriptions for relevant families of relation mappings would simplify both the analysis and the implementation of separation routines. Finally, embedding the proposed cutting-plane techniques into an adaptive piecewise-linear solving procedure, where the discretization is refined dynamically based on solution information, constitutes a promising avenue for combining the strengths of polyhedral tightening with adaptive approximation strategies.

References

[1] Nilanjan Adhya, Mohit Tawarmalani, and Nikolaos V. Sahinidis. A lagrangian approach to the pooling problem. *Industrial & Engineering Chemistry Research*, 38(5):1956–1972, 1999.

[2] Pietro Belotti. Disjunctive cuts for nonconvex minlp. In Jon Lee and Sven Leyffer, editors, *Mixed Integer Nonlinear Programming*, pages 117–144, New York, NY, 2012. Springer New York.

[3] Kristin Braun and Robert Burlacu. Benchmarking piecewise linear reformulations for minlps: A computational study based on the open-source framework pwl-t-rex, September 2025.

[4] Robert Burlacu, Geißler Björn, and Lars Schewe. Solving mixed-integer nonlinear programmes using adaptively refined mixed-integer linear programmes. *Optimization Methods and Software*, 35(1):37–64, January 2020.

[5] Robert Burlacu, Patrick Gemander, and Tobias Kuen. The Bipartite Implication Polytope: Conditional Relations over Multiple Sets of Binary Variables, April 2024.

[6] Joey Huchette and Juan Pablo Vielma. Nonconvex piecewise linear functions: Advanced formulations and simple modeling tools. *Operations Research*, 71(5):1835–1856, 2023.

[7] R. G. Jeroslow and J. K. Lowe. *Modelling with integer variables*, pages 167–184. Springer Berlin Heidelberg, Berlin, Heidelberg, 1984.

[8] Anil B. Keha, Daniela de Farias, and George L. Nemhauser. A branch-and-cut algorithm without binary variables for nonconvex piecewise linear optimization. *Operations Research*, 54(5):847–858, 2006.

[9] J. Kim et al. Learning symbolic expressions: Mixed-integer formulations, cuts, and heuristics. *arXiv*, 2021.

[10] Bochuan Lyu, Illya V. Hicks, and Joey Huchette. Modeling combinatorial disjunctive constraints via junction trees. *Mathematical Programming*, 204(1):385–413, March 2024.

[11] Harry M. Markowitz and Alan S. Manne. On the Solution of Discrete Programming Problems. *Econometrica*, 25(1):84–110, 1957. Publisher: [Wiley, Econometric Society].

[12] Garth P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I — Convex underestimating problems. *Mathematical Programming*, 10(1):147–175, 1976.

[13] Maximilian Merkert. *Solving Mixed-Integer Linear and Nonlinear Network Optimization Problems by Local Reformulations and Relaxations.* PhD thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2015.

[14] Harsha Nagarajan, Mowen Lu, Emre Yamangil, and Russell Bent. Tightening McCormick Relaxations for Nonlinear Programs via Dynamic Multivariate Partitioning. pages 369–387, 2016.

[15] Ivo Nowak. *Relaxation and Decomposition Methods for Mixed Integer Nonlinear Programming*, volume 152 of *International Series of Numerical Mathematics.* Birkhäuser, Basel, 2005.

[16] Anureet Saxena, Pierre Bonami, and Jon Lee. Disjunctive cuts for non-convex mixed integer quadratically constrained programs. In Andrea Lodi, Alessandro Panconesi, and Giovanni Rinaldi, editors, *Integer Programming and Combinatorial Optimization*, pages 17–33, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[17] Artur M. Schweidtmann, Dominik Bongartz, Daniel Grothe, Tim Kerkenhoff, Xiaopeng Lin, Jaromil Najman, and Alexander Mitsos. Deterministic global optimization with Gaussian processes embedded. *Mathematical Programming Computation*, 13(3):553–581, 2021.

[18] Juan Pablo Vielma. Mixed integer linear programming formulation techniques. *SIAM Review*, 57(1):3–57, 2015.

[19] Juan Pablo Vielma, Shabbir Ahmed, and George L Nemhauser. Mixed-integer models for nonseparable piecewise-linear optimization: Unifying framework and extensions. *Operations Research*, 58(2):303–315, 2010.

[20] Juan Pablo Vielma, Anil B. Keha, and George L. Nemhauser. Nonconvex, lower semicontinuous piecewise linear optimization. *Discrete Optimization*, 5(2):467–488, 2008.

[21] Stefan Vigerske et al. MINLPLib: A library of mixed-integer and continuous nonlinear programming instances. `https://www.minlplib.org/`, 2026. Accessed: 2026-02-27.

[22] Yufei Zhao and Daniela de Farias. The piecewise linear optimization polytope: New inequalities and intersection with semi-continuous constraints. *Mathematical Programming*, 141(1-2):209–240, 2013.

A Proof of Theorem 2

Let

$$\sum_{i\in[\gamma]}\sum_{j\in[\alpha]} a^i_j x^i_j \leq \sum_{l\in[\beta]} b_l y_l + c, \tag{16}$$

be the facet defining inequality for facet F of $P(\varphi)$. Since we work with rational polytopes, without loss of generality, we assume the coefficients a^i_j, b_l and c to be integer.

Case 1: Suppose for an implying set $i \in [\gamma]$, there exists an index $j' \in [\alpha]$ such that for every vertex $\mathbf{v_j} \in F$, it holds true that $j_i \neq j'$. This means $x^i_{j'} = 0$ for every vertex $\mathbf{v_j} \in F$. Therefore, it follows that F is contained in the hyperplane

$$H = \{(\mathbf{x}^1, \dots, \mathbf{x}^\gamma, \mathbf{y}) \in \mathbb{R}^{\alpha\gamma+\beta} : x^i_{j'} = 0\}.$$

Since we know the structure of vertices, there exists a vertex $\mathbf{v_j} \in P(\varphi) \setminus F$ with $j_i = j'$, and it holds $P(\varphi) \nsubseteq H$. Because of the facet-property of F, it is then $F = H \cap P(\varphi)$. This means, that F is induced by the lower bound $x^i_{j'} \geq 0$.

Suppose analogously for the implied set, there exists $l \in [\beta]$ such that for every vertex $\mathbf{v_j} \in F$ holds $\varphi(\mathbf{j}) \neq l$. Then, facet F lies in the hyperplane

$$H = \{(\mathbf{x}^1, \dots, \mathbf{x}^\gamma, \mathbf{y}) \in \mathbb{R}^{\alpha\gamma+\beta} : y_l = 0\}.$$

Since φ is surjective, there exists a vertex $\mathbf{v_j} \in P(\varphi) \setminus F$ with $\varphi(\mathbf{j}) = l$. Therefore, it is $P(\varphi) \nsubseteq H$ and F is induced by the lower bound $y_l \geq 0$.

Case 2: Now, excluding the previous cases, we show that properties (a) and (b) from Definition 1 are true for the facet defining inequality (16).

Property (a)**:** Since inequality (16) is valid, for any vertex $\mathbf{v_j}$ of $P(\varphi)$, $\mathbf{j} \in [\alpha]^\gamma$, it holds that

$$b_{\varphi(\mathbf{j})} \geq \sum_{i\in[\gamma]} a^i_{j_i} - c.$$

Consider $l' \in [\beta]$ arbitrary but fixed. Because φ is surjective, there exists at least one $\mathbf{j} \in [\alpha]^\gamma$ with $\varphi(\mathbf{j}) = l'$. It is

$$b_{l'} \geq \sum_{i\in[\gamma]} a^i_{j_i} - c \quad \text{for all} \quad \mathbf{j} \in [\alpha]^\gamma, \quad \text{where} \quad \varphi(\mathbf{j}) = l'.$$

Since we exclude the upper case, there exists at least one $\mathbf{v_j} \in F$ with $\varphi(\mathbf{j}) = l'$. For all such vertices, inequality (16) holds with equality, i.e.,

$$b_{l'} = \sum_{i\in[\gamma]} a^i_{j_i} - c.$$

Therefore, we conclude that property (a) holds.

Property (b)**:** First, we construct an inequality equivalent to (16), that takes the form

$$\sum_{i\in[\gamma]} \sum_{j\in[\alpha]} \underbrace{(a^i_j - a^i_{\max} + b_{\max} - b_{\min})}_{=:\hat{a}^i_j} x^i_j \leq \sum_{l\in[\beta]} \underbrace{(b_l - b_{\min})}_{=:\hat{b}_l} y_l + \underbrace{(\gamma - 1)(b_{\max} - b_{\min})}_{=:\hat{c}}, \tag{17}$$

and which construction we explain in the following.

To achieve the equivalent form (17), we "align" all maximum coefficients at $b_{\max} - b_{\min}$. For the left hand side, we add the following scaled multiple choice constraint to (16),

$$(-a^i_{\max} + b_{\max} - b_{\min}) \sum_{j\in[\alpha]} x^i_j = -a^i_{\max} + b_{\max} - b_{\min}, \tag{18}$$

for every $i \in [\gamma]$. Then, for all $i \in [\gamma]$, the maximum coefficient $\hat{a}^i_{\max}$ in (17) satisfies

$$\hat{a}^i_{\max} = a^i_{\max} - a^i_{\max} + b_{\max} - b_{\min} = b_{\max} - b_{\min}. \tag{19}$$

Further, for the left hand side, we add the following scaled multiple choice constraint to (16)

$$0 = -b_{\min} \sum_{l\in[\beta]} y_l + b_{\min}. \tag{20}$$

Then, for the maximum coefficient $\hat{b}_{\max}$ in (17), it is $\hat{b}_{\max} = b_{\max} - b_{\min}$. Note, that for all $l \in [\beta]$ it holds

$$\hat{b}_l = b_l - b_{\min} \geq 0. \tag{21}$$

Constant $\hat{c}$ in (17) we compute by gathering all constants that we add with (18) for $i \in [\gamma]$ and with (20) to (16), i.e.,

$$\begin{aligned}
\hat{c} &= c + \sum_{i\in[\gamma]} (-a^i_{\max} + b_{\max} - b_{\min}) + b_{\min} \\
&= c - \sum_{i\in[\gamma]} a^i_{\max} + \gamma(b_{\max} - b_{\min}) + b_{\min} \\
&= c - \sum_{i\in[\gamma]} a^i_{\max} + b_{\max} + (\gamma - 1)(b_{\max} - b_{\min}) \\
&\overset{\text{Lemma 2}}{=} (\gamma - 1)(b_{\max} - b_{\min}).
\end{aligned}$$

Then, we obtain inequality (17), that is equivalent to (16) and therefore, also facet defining. So, we have clarified the derivation of (17).

Now, due to property (a) on (17), for every $l \in [\beta]$, it holds

$$\hat{b}_l = \max_{\mathbf{j}\in[\alpha]^\gamma, \varphi(\mathbf{j})=l} \left\{ \sum_{i\in[\gamma]} \hat{a}^i_j - \hat{c} \right\}. \tag{22}$$

We define another inequality,

$$\sum_{i\in[\gamma]}\sum_{j\in[\alpha]}\tilde{a}^i_j x^i_j \leq \sum_{l\in[\beta]}\hat{b}_l y_l + \hat{c}, \tag{23}$$

with new coefficients $\tilde{a}^i_j := \max\{0, \hat{a}^i_j\}$ for $i \in [\gamma]$, $j \in [\alpha]$. For all $i \in [\gamma]$ and $j \in [\alpha]$, by definition it is

$$\tilde{a}^i_j \geq \hat{a}^i_j = a^i_j - a^i_{\max} + b_{\max} - b_{\min},$$

which means that (23) dominates (17).

Assume property (b) does not hold for (16), i.e., for at least one $i \in [\gamma]$, it holds

$$\hat{a}^i_{\min} = a^i_{\min} - a^i_{\max} + b_{\max} - b_{\min} < 0.$$

Consequently, inequality (23) strictly dominates inequality (17). In addition to strict dominance, if we proof the validity of inequality (23) for $P(\varphi)$, this would contradict the facet defining property of (17), and we have proven property (b).

Assume inequality (23) is not valid for $P(\varphi)$. Then, leveraging the characterization in Lemma 1 there exists $l' \in [\beta]$ and $\mathbf{j'} \in [\alpha]^\gamma$ such that

$$\sum_{i\in[\gamma]}\hat{a}^i_{j'} - \hat{c} \overset{(22)}{=} \hat{b}_{l'} < \sum_{i\in[\gamma]}\tilde{a}^i_{j'} - \hat{c} = \max_{\mathbf{j}\in[\alpha]^\gamma,\varphi(\mathbf{j})=l'}\left\{\sum_{i\in[\gamma]}\tilde{a}^i_j - \hat{c}\right\},$$

resulting in

$$\sum_{i\in[\gamma]}\hat{a}^i_{j'_i} < \sum_{i\in[\gamma]}\tilde{a}^i_{j'_i}.$$

Due to the integrality of all coefficients in (16), we collect in index set $I \subset [\gamma]$ all indices $i \in [\gamma]$ for which it holds

$$a^i_{j'_i} - a^i_{\max} + b_{\max} - b_{\min} \leq -1. \tag{24}$$

Then, it is $a^i_{j'_i} - a^i_{\max} \leq b_{\max} - b_{\min}$ for the other $i \in [\gamma] \setminus I$. We conclude

$$\begin{aligned}\sum_{i\in[\gamma]}\hat{a}^i_{j'_i} &= \sum_{i\in[\gamma]}(a^i_{j'_i} - a^i_{\max} + b_{\max} - b_{\min})\\ &\overset{(24)}{\leq} -|I| + \sum_{i\in[\gamma]\setminus I}(a^i_{j'_i} - a^i_{\max} + b_{\max} - b_{\min})\\ &\overset{(19)}{\leq} -|I| + (\gamma - |I|)(b_{\max} - b_{\min})\\ &< (\gamma - |I|)(b_{\max} - b_{\min})\\ &\leq (\gamma - 1)(b_{\max} - b_{\min}) = \hat{c}.\end{aligned} \tag{25}$$

From equality (22), it follows

$$\hat{b}_{l'} = \sum_{i\in[\gamma]}\hat{a}^i_{j'} - \hat{c} \overset{(25)}{<} \hat{c} - \hat{c} = 0,$$

which is a contradiction to (21). This means, that inequality (23) is valid for $P(\varphi)$ and property (b) holds for all $i \in [\gamma]$ for inequality (16).