The Bipartite Implication Polytope: Conditional Relations over Multiple Sets of Binary Variables

Robert Burlacu* Patrick Gemander† Tobias Kuen‡

June 17, 2026

Abstract

Inspired by its occurrence as a substructure in a stochastic railway timetabling model, we study in this work a special case of the bipartite boolean quadric polytope. It models conditional relations across three sets of binary variables, where selections within two implying sets imply a choice in a corresponding implied set. We call this polytope the *Bipartite Implication Polytope*.

We introduce a new class of valid inequalities and prove that, in contrast to the well-known McCormick inequalities, they are sufficient to completely characterize the description of the polytope. We develop a separation algorithm that finds these inequalities in polynomial time and propose an additional clique-based method for precomputing tight cuts. Furthermore, we show that for a chain of several bipartite implication relations, the descriptions of the bipartite implication polytopes for each individual relation already yield the convex hull of the chained polytope. This is present in our application from the field of stochastic timetabling and also enables a broader application of our results in practice. A comprehensive computational study shows the usefulness of the new inequalities in state-of-the-art branch-and-cut solvers for real-world timetabling applications and instances of the quadratic assignment problem.

Keywords: Quadratic Assignment Problem, Integer Programming, Fixed Recourse Stochastic Problem, Boolean Quadric Polytope, Bipartite Graphs, Multiple-Choice Constraints, Convex Hull, Branch-and-Cut, Railway Timetabling
MSC Codes: 90C09, 90C20, 90C25, 90C27, 90C35, 90C57, 90C90

1 Introduction

The famous boolean quadric polytope

$$\mathrm{QP}(G) := \mathrm{conv}\left\{(x,z) \in \{0,1\}^{|V|+|E|} \mid x_i x_j = z_{ij},\ (ij) \in E\right\}$$

was introduced in [35] for general undirected graphs $G = (V, E)$. In this paper, we consider the case, where $G = (X \cup Y, E)$ is bipartite and additional multiple-choice constraints apply to both sets X and Y. This structure is inherent in diverse optimization problems, for instance where bipartite graphs serve as a modeling basis, as in assignment and transportation problems, and additionally a single option must be selected from a large number of alternatives.

For illustration purposes, consider the search for the shortest path in a time-expanded graph, where the nodes have three attributes: time, velocity, and position. Such a graph is employed to minimize the energy consumption of a train's driving profile [47]. Here, the polytope introduced in this work does not appear only once, but multiple times as a substructure, namely for each bipartite

*Fraunhofer-Institut für Integrierte Schaltungen IIS, Gruppe Optimization, Nordostpark 84, 90411 Nürnberg, Germany. E-mail: robert.burlacu@iis.fraunhofer.de
†E-mail: patrick.gemander@fau.de
‡E-mail: tobias.kuen@iis.fraunhofer.de

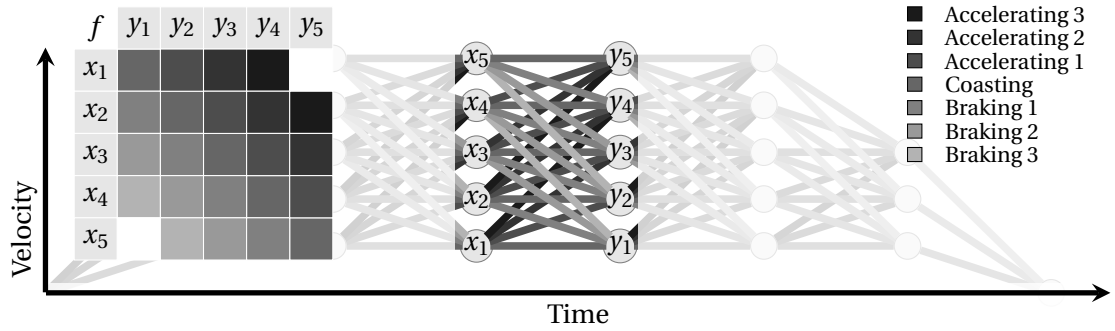


Figure 1: Example time-expanded graph with grouped edges.

subgraph with nodes at timestamps $t, t+1$ for $t \in \{1, ..., T-1\}$ where T is the number of timestamps considered in the time-expanded graph. Every subgraph that is formed by considering all nodes between two consecutive timestamps exhibits a bipartite structure. The edges within these subgraphs are assigned costs that indicate the energy consumption of the train during the travel between the two timestamps.

The example in Figure 1 neglects the position axis and only considers time and velocity for illustrative reasons. More formally, we can represent the shortest path problem as a binary quadratic program. To this end, at timestamp i, we associate each node $u \in U$ with a variable x_u and at timestamp $i+1$, each node $v \in V$ with a variable y_v. For each edge $(uv) \in E$, we introduce a variable p_{uv} with assigned costs c_{uv}. For each point in time, we have to decide for a specific velocity and position, which implies a multiple-choice constraint at both observed timestamps. Consequently, the objective is given by

$$\min \left\{ \sum_{(uv)\in E} c_{uv} p_{uv} \mid \sum_{u\in U} x_u = \sum_{v\in V} y_v = 1,\ x_u y_v = p_{uv},\ (uv) \in E,\ (x,y,p) \in \{0,1\}^{|U|+|V|+|E|} \right\}.$$

In practice, the specific edge that is chosen can be inferred from the selected node variables, so explicit edge variables are not needed. Instead of including a variable p_{uv} for each edge (u, v), we can group edges that share the same cost c_l and introduce a single variable z_l for each cost group $l \in L$. The variable z_l thus represents the selection of any edge within group l, and the corresponding cost c_l enters the objective function only once. This reformulation can substantially reduce the number of variables compared to traditional edge-based linear models for shortest path problems, which may be advantageous in some applications. In the example of Figure 1, these cost groups correspond to three different levels of braking, coasting and three levels of accelerating. With $f : E \to L$ as the function that maps each edge to its group, we can now formulate the problem as

$$\min \left\{ \sum_{l\in L} c_l z_l \mid \sum_{u\in U} x_u = \sum_{v\in V} y_v = \sum_{l\in L} z_l = 1,\ x_u y_v \le z_{f(uv)},\ (uv) \in E,\ (x,y,z) \in \{0,1\}^{|U|+|V|+|L|} \right\}.$$

Note that the multiple-choice constraint on the cost variables z is not directly implied by the multiple-choice constraints on the node variables x and y but has to be added since exactly one edge is active per timestamp. This formulation gives rise to a distinctive polytope, termed the *Bipartite Implication Polytope (BIP)*, because it entails the selection of one variable each from two *implying sets* of binary variables, which in turn implies the selection of one binary variable from the *implied set*.

Potential Applications Potential applications for the BIP are manifold. One natural candidate emerges in the field of fixed-recourse stochastic programming, which deals with optimization problems involving decision-making under uncertainty. A subclass of these problems - namely those

with endogenous uncertainties - deals with uncertainties that depend on the decisions made and optimized. When modeling uncertain outcomes using scenario variables, these variables must be coupled to the decision variables of the problem. In a planning problem, an initial decision for an event $\mathscr{B}$ exists, along with a realization of that event, which may be affected by the outcome of another event, $\mathscr{A}$. The decisions and realizations of both events are modeled as four distinct sets of binary variables, each governed by a multiple-choice constraint. Three of the four sets have a bipartite implication relation. First, the Decision Set B, which represents the planned choice for $\mathscr{B}$, let this be the set $\{b_1,...,b_m\}$. Second, the Realization Set A', which represents the actual outcome or realization of $\mathscr{A}$, let this be the set $\{a'_1,...,a'_n\}$. And third, the Realization Set B', which represents the actual outcome or realization of $\mathscr{B}$, let this be the set $\{b'_1,...,b'_p\}$. The actual realization of $\mathscr{B}$, the choice from B', is jointly determined by both the planned decision for $\mathscr{B}$, the choice from B, and the actual realization of $\mathscr{A}$, the choice from A'. The BIP models this exact scenario. In this context, the implying sets are the decision set B and the realization set A', and the implied set is the resulting realization set B'. The BIP formulation then enforces the logical relationship: "IF decision b_i is planned AND realization a'_j is observed, THEN realization b'_k must occur."

A concrete example of an application with endogenous uncertainties is a stochastic railway timetabling model, which is one of the main motivations for this paper and is described in [7]. The underlying clique problem with multiple-choice constraints was introduced in [10] and analyzed in [7]. In [9], the scenario extension was added, where the delay of a train is an uncertain value, depending on decisions regarding departure and running times.

Another occurrence of the BIP can be found in the quadratic assignment problem (QAP). It poses a fundamental optimization challenge that has intrigued researchers and practitioners across various disciplines. Originating in operations research, the QAP involves optimizing the allocation of resources considering both assignment and distance-related costs, presenting a significant computational challenge. The QAP finds broad applications in diverse fields. First, it was introduced by [26] in the context of optimally locating facilities. Other applications include scheduling problems ([17]), airline maintenance operations ([32]) or reactionary chemistry ([44]). A comprehensive overview of the QAP is given in [5]. In the quadratic integer formulation, costs are assigned to products of binary variables that are present in several multiple-choice constraints. Similar to the above mentioned shortest path problem in time-expanded graphs, we can group products of variables with equal costs and with that establish a bipartite implication substructure. While we do not aim to compete with specialized QAP solvers, this well-studied problem with its standard benchmark library QAPLIB [6] provides an excellent testbed to validate the performance of our developed cuts on problems with multiple chained bipartite implication substructures, indicating their potential broader applicability. Further applications in the fields of approximations and relaxations for non-linear programming and the integration of Bayesian networks into mixed-integer programs are discussed in detail later in this work.

Related Literature The foundational work in [35], introducing the boolean quadric polytope QP(G) for general undirected graphs G, has been pivotal, laying the groundwork for a deeper understanding of unconstrained binary quadratic programming. Although no constraints are involved in these optimization problems, the quadratic objective alone yields an NP-hard problem, as shown in [2]. Over the last decades, the boolean quadric polytope has been studied intensively, resulting in many facet classes and corresponding separation algorithms, and the observation of symmetries and other geometric properties; see e.g. [3, 27, 40]. We refer the reader to [25] for a comprehensive survey on applications and solution methods for general unconstrained binary quadratic programming. In recent years, the geometry and other properties of the bipartite boolean quadric polytope BQP(G), the special case of QP(G) where G is bipartite, have been studied in [37, 38, 42, 43] together with various heuristic approaches ([15, 20, 23, 45]). Applications containing this polytope stem, for example, in the fields of data mining [30] and bioinformatics [11].

Binary quadratic programs with linear and/or quadratic constraints are among the best studied

classes of integer nonlinear problems, primarily because they allow to model a large number of diverse applications [4]. Although a variety of different solution approaches have been proposed over the last decades, these programs are usually tackled by linearizing the quadratic parts of the problem and subsequently passing the equivalent linear representation to a general-purpose mixed-integer linear programming solver. Two of the most commonly used linearization schemes are the so-called standard linearization from [19] and Glover's method from [18]. Another frequently utilized approach is proposed in [39]. Here, the authors introduced a more compact but less tight linearization formulation and showed its performance advantage on knapsack-constrained instances. Recently, the authors of [16] conducted a comprehensive computational study on various applications to determine the optimal manner of applying these linearization methods with additional enhancements. Alongside these general methods, a wide range of approaches have been developed that are specifically tailored to different classes of constraints. For example, in [28] a compact reformulation for binary quadratic programs with assignment constraints has been proposed. A thorough comparison of different methods for binary quadratic programs with an additional cardinality constraint is given in [29]. In recent years, multiple-choice (or set-packing) structures have been studied in greater depth, particularly in relation to the bipartite boolean quadric polytope (BQP). [8] investigated a specific case of the BQP, incorporating multiple-choice constraints applied exclusively to partitions of the nodes V of a bipartite graph. This approach was motivated by applications such as pooling problems with fixed input proportions, where a partition on the V set restricts selection to at most one node per subset. The authors identified various facet classes and facet-preserving operations pertinent to this structure, introduced new facet classes arising from the multiple-choice constraints, and developed lifting operations to extend these facets. Additionally, they find various cases in which the polytope is completely described via the relaxation-linearization inequalities. In contrast, in this paper, we consider two multiple-choice equalities for the nodes in the bipartite graph, one for each part. We also present a hierarchical characterization for the convex hull of the feasible points and add a graphical interpretation for the discovered inequalities that enhances the understanding of the complex problem class. The bipartite quadratic assignment problem [36] and the bilinear assignment problem [48] are also closely related problems that involve the study of BQP(G) with multiple-choice constraints on multiple, non-disjoint subsets of both X and Y. In these studies, the authors focus more on complexity analysis for optimizing over the specific polytopes and provide heuristics to speed up the solving process.

Contribution Initially motivated by an application from real-world stochastic timetabling, we study a polyhedral substructure of this problem that models conditional relations across three sets of binary variables, i.e., where selections within two implying sets imply a choice in a corresponding implied set: the BIP. Our contribution is a new class of valid inequalities for this polytope. In contrast to the unconstrained (bipartite) boolean quadric polytope, the special structure of the BIP allows us to prove that this class of inequalities is sufficient for a complete description. We develop a separation algorithm that finds these inequalities in polynomial time. Supplementary to this, we present a clique-based method that is able to determine a priori cuts that are helpful for the solving process. Furthermore, we show that for set descriptions that consist of multiple BIPs which share variable sets, the descriptions of the individual BIPs already provide a complete description of the whole polytope. This enables a much broader application of our results in practice. In a comprehensive computational study, we investigate the aforementioned applications from the field of real-world stochastic timetabling and the quadratic assignment problem. We demonstrate the strength of the new cuts by incorporating them into the state-of-the-art solver Gurobi [21], which speeds up the solution process by orders of magnitude.

Structure of the Paper After a short definition of the BIP in Section 2, we derive a new class of valid inequalities in Section 3. We then prove in Section 4 that these inequalities together with bound inequalities completely describe the BIP. Additionally, we present efficient ways to use the defined

inequalities to optimize over the BIP using either a precomputation routine or a separation algorithm. Preparing the comprehensive computational study of Section 6, we first analyze the chaining of multiple BIPs in Section 5, that arise in the application for stochastic railway timetabling. Finally, we explore other potential areas of application for future research in Section 7, including approximations and relaxations for nonlinear programming, as well as the integration of Bayesian networks into mixed-integer programs.

2 Problem Definition

Let $x \in \{0,1\}^\alpha$, $y \in \{0,1\}^\beta$, and $z \in \{0,1\}^\gamma$ be three vectors of binary variables and $\alpha,\ \beta,\ \gamma \geq 1$. The implications between the three vectors are given by a relation matrix M. We assume that each combination of implying variables implies a corresponding element in the implied set, otherwise we can introduce a variable for infeasible combinations and set it equal to 0. We denote the set $\{1,...,\alpha\}$ as $[\alpha]$. If $x_i = 1$ holds for some $i \in [\alpha]$, and $y_j = 1$ for some $j \in [\beta]$, this implies the choice $z_l = 1$, where $l = M_{ij}$ is the corresponding entry of the implication relation matrix. Note that we assume that each $l \in [\gamma]$ is contained in M. We must choose exactly one x-, one y-, and one z-variable to be equal to one, while respecting the implications stated in M. The set of feasible points is thus given by:

$$S(M) := \left\{ (x,y,z) \in \{0,1\}^{\alpha+\beta+\gamma} \mid x_i \cdot y_j \leq z_{M_{ij}}\ \forall (i,j) \in [\alpha] \times [\beta],\ \sum_{i=1}^{\alpha} x_i = \sum_{j=1}^{\beta} y_j = \sum_{l=1}^{\gamma} z_l = 1 \right\}.$$

We can linearize the bilinear terms in the definition of $S(M)$ to equivalently write:

$$S(M) = \left\{ (x,y,z) \in \{0,1\}^{\alpha+\beta+\gamma} \mid x_i + y_j \leq z_{M_{ij}} + 1\ \forall (i,j) \in [\alpha] \times [\beta],\ \sum_{i=1}^{\alpha} x_i = \sum_{j=1}^{\beta} y_j = \sum_{l=1}^{\gamma} z_l = 1 \right\}.$$

In the following, we consider the so-called *bipartite implication polytope* $BIP(M) := \text{conv}(S(M))$, which arises as the convex hull of $S(M)$. The multiple-choice equations imply that the polytope is not full dimensional.

Observation 1. We have $\dim(BIP(M)) \leq \alpha + \beta + \gamma - 3$.

Note that there are cases of M for which $\dim(BIP(M)) < \alpha + \beta + \gamma - 3$ holds. For example, if:

$$M = \begin{pmatrix} 1 & 2 & 2 \\ 3 & 1 & 1 \\ 3 & 1 & 1 \end{pmatrix},$$

the equation $x_1 + y_2 + y_3 = z_1 + 2z_2$ is valid for $BIP(M)$, in addition to the multiple-choice constraints.

Any optimization problem over $BIP(M)$ is inherently easy and can be solved in polynomial time just by enumerating all the vertices.

Lemma 1. *The vertices of $BIP(M)$ are given by $e_i + e_{\alpha+j} + e_{\alpha+\beta+M_{ij}}$ for all $i \in [\alpha]$ and $j \in [\beta]$, where e_m for $m \in [\alpha+\beta+\gamma]$ denotes the m-th standard unit vector in $\{0,1\}^{\alpha+\beta+\gamma}$.*

Proof. As $BIP(M)$ is the convex hull of a set of binary points, these are precisely the vertices of $BIP(M)$. □

It can still be beneficial to study the facet description of $BIP(M)$ whenever there are applications in which the determined constraints are part of a larger system.

3 Valid Inequalities

In this section we describe and fully characterize a new class of valid inequalities for $BIP(M)$ which we call n-block inequalities because of their block-like representation in the relation matrix M.

3.1 n-Block Inequalities

A *block* $M_{X,Y}$ is defined as the submatrix of M where the selected row indices are in $X \subseteq [\alpha]$ and the selected column indices are in $Y \subseteq [\beta]$. The consideration of such blocks allows to strengthen the formulation of $BIP(M)$. We denote the set of z-indices contained in the block $M_{X,Y}$ by

$$Z^M(X,Y) \coloneqq \{M_{ij} : (i,j) \in X \times Y\}$$

If we have $x_i = 1$ and $y_j = 1$ with $i \in X$ and $j \in Y$, then it follows that $z_l = 1$ for some $l \in Z^M(X,Y)$. Consequently, we can formulate the following *1-block-inequality*

$$\sum_{i \in X} x_i + \sum_{j \in Y} y_j \leq \sum_{l \in Z^M(X,Y)} z_l + 1, \tag{1}$$

that is valid for $BIP(M)$ for each subset of rows $X \subseteq [\alpha]$ and each subset of columns $Y \subseteq [\beta]$.

Observation 2. There exist at most $(2^\alpha - 1)(2^\beta - 1)$ many non-equivalent up to scaling 1-block inequalities that are valid for $BIP(M)$.

We can derive even stronger inequalities when taking $n \in \mathbb{N}$ blocks into account. For each $k \in [n]$, select rows $X_k \subseteq [\alpha]$ and columns $Y_k \subseteq [\beta]$ of the matrix M to define n blocks such that the subsets are sorted by inclusion as follows:

$$X_{k+1} \subseteq X_k \quad \text{and} \quad Y_k \subseteq Y_{k+1} \quad \text{for all} \quad k \in [n-1]. \tag{2}$$

For a subset of the chosen blocks, indexed by $K \subseteq [n]$, we define the set of entries of M that are located in the intersection of κ-many of the blocks as

$$\Xi_\kappa^M(K) \coloneqq \left\{M_{ij} : \left|\left\{k \in K : (i,j) \in X_k \times Y_k\right\}\right| \geq \kappa\right\}. \tag{3}$$

Then we can construct what we call the *n-block inequality*

$$\sum_{i \in [\alpha]} a_i x_i + \sum_{j \in [\beta]} b_j y_j \leq \sum_{l \in [\gamma]} c_l z_l + n, \tag{4}$$

where the respective variable coefficients are given by

$$a_i = |\{k \in [n] : i \in X_k\}|, \quad i \in [\alpha], \tag{5}$$

the number of blocks row i is included,

$$b_j = \left|\left\{k \in [n] : j \in Y_k\right\}\right|, \quad j \in [\beta], \tag{6}$$

the number of blocks column j is included and

$$c_l = \max\left\{\kappa \in [n] : l \in \Xi_\kappa^M([n])\right\}, \quad l \in [\gamma], \tag{7}$$

the maximum number of intersecting blocks at a cell with entry l.

We can use the sorting of the blocks by inclusion to efficiently determine the number of blocks intersecting in one cell (i,j) of M from the coefficients a_i and b_j. The value of a_i indicates that i is contained in the first a_i blocks M_{X_k,Y_k} for $k \in [a_i]$, whereas b_j indicates that j is contained in the last b_j blocks M_{X_k,Y_k} for $k \in \{n - b_j + 1, \ldots, n\}$.

Therefore, the indices blocks intersecting in cell (i,j) are $\{a_i + b_j - n, ..., a_i\}$ if $a_i \geq a_i + b_j - n$, otherwise the cut is empty. The number of intersecting blocks in (i,j) is $\max\{0, a_i + b_j - n\}$.

The coefficient of an element $l \in [\gamma]$ can thus be calculated as

$$c_l = \max_{i,j \in [\alpha]\times[\beta]: M_{ij}=l} \max\{0, a_i + b_j - n\} \quad \forall l \in [\gamma]. \tag{8}$$

We will use this formula later in this section when characterizing n-block inequalities. Note that the n blocks are an indexed collection, allowing for repetition; the same block defined by identical subsets X_k and Y_k can appear multiple times for different k. Consequently, n is not bounded by the number of distinct blocks that can be formed. In fact, we observe this stacking of identical blocks in facet-defining inequalities.

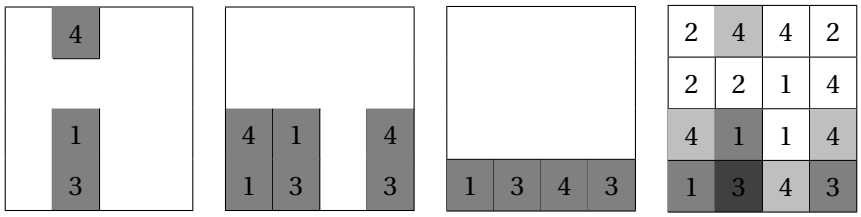
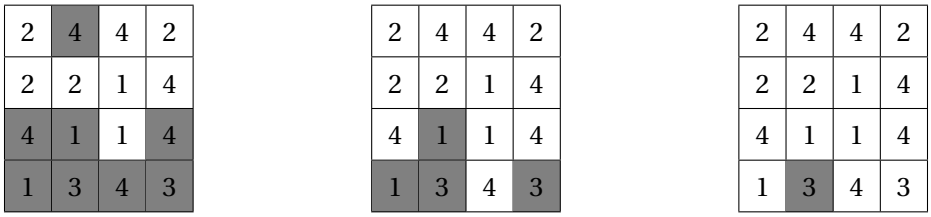


Figure 2: Construction of the 3-block inequality in Example 1.



(a) $\Xi_1^M([3]) = \{1,3,4\}$ (b) $\Xi_2^M([3]) = \{1,3\}$ (c) $\Xi_3^M([3]) = \{3\}$

Figure 3: Set of entries of M that are located in the intersection of $\kappa = 1,2,3$-many of all 3 blocks of the 3-block inequality in Example 1

.

Example 1. Figure 2 illustrates the construction of the 3-block inequality

$$x_1 + 2x_3 + 3x_4 + 2y_1 + 3y_2 + y_3 + 2y_4 \leq 2z_1 + 3z_3 + z_4 + 3$$

out of the three blocks $M_{\{1,3,4\},\{2\}}$, $M_{\{3,4\},\{1,2,4\}}$ and $M_{\{4\},\{1,2,3,4\}}$ of the matrix

$$M = \begin{pmatrix} 2 & 4 & 4 & 2 \\ 2 & 2 & 1 & 4 \\ 4 & 1 & 1 & 4 \\ 1 & 3 & 4 & 3 \end{pmatrix}.$$

The color of each cell signifies the number of blocks intersecting there. For each $l \in [4]$, the color of the darkest cell it is contained in indicates the maximum value of κ for which l is in $\Xi_\kappa^M(\{1,2,3\})$ as depicted in Figure 3. This value corresponds to its coefficient c_l. We can derive the color of a given cell (i,j) in the matrix M efficiently from the coefficients a_i and b_j via the previously defined sorting of the blocks by inclusion. If we take for example $(i,j) = (4,1)$, where $a_4 = 3$ and $b_1 = 2$ hold, we know that row 4 is in the two leftmost and column 1 in the two rightmost of the three blocks depicted in Figure 2. This implies that they jointly only lie in the two rightmost blocks, which is why $(4,1)$ is in the intersection of exactly 2 blocks. Note that this 3-block inequality dominates the sum of the 1-block inequalities derived when considering each block individually, because some of the coefficients of

the z-variables are smaller. For example, $l = 4$ does not lie in the intersection of any two of the three blocks, but is contained in each of them. Therefore $c_4 = 1$, whereas in the addition of the three 1-block inequalities the coefficient of z_4 would be 3.

Lemma 2. *The n-block inequalities* (4) *are valid for $BIP(M)$ for all $n \in \mathbb{N}$.*

Proof. We prove the result by induction over the number of blocks n. For $n = 1$, the validity of the 1-block inequalities follows from construction.

For the case $n = 2$, we prove the validity of the 2-block inequalities obtained from two blocks M_{X_1,Y_1} and M_{X_2,Y_2}. To this end, we sum up the two 1-block inequalities for the two blocks (9a), (9b), the 1-block inequality (9c) for the intersection $M_{X_1\cap X_2,Y_1\cap Y_2}$ and the inequalities (9d) and (9e) derived by adding the multiple-choice constraints for the x- and y-variables and non-negativity constraints for some of the z-variables, respectively:

$$\sum_{i\in X_1} x_i + \sum_{j\in Y_1} y_j - \sum_{l\in Z^M(X_1,Y_1)} z_l \leq 1 \tag{9a}$$

$$+ \quad \sum_{i\in X_2} x_i + \sum_{j\in Y_2} y_j - \sum_{l\in Z^M(X_2,Y_2)} z_l \leq 1 \tag{9b}$$

$$+ \quad \sum_{i\in X_1\cap X_2} x_i + \sum_{j\in Y_1\cap Y_2} y_j - \sum_{l\in Z^M(X_1\cap X_2,Y_1\cap Y_2)} z_l \leq 1 \tag{9c}$$

$$+ \quad \sum_{i\in[\alpha]} x_i + \sum_{j\in[\beta]} y_j \leq 2 \tag{9d}$$

$$+ \quad \sum_{l\in Z^M(X_1\cap X_2,Y_1\cap Y_2)\cup\left(\left(Z^M(X_1,Y_1)\cup Z^M(X_2,Y_2)\right)\setminus\left(Z^M(X_1,Y_1)\cap Z^M(X_2,Y_2)\right)\right)} -z_l \leq 0 \tag{9e}$$

$$= \quad 2\cdot\left(\sum_{k=1}^{2}\sum_{i\in X_k} x_i + \sum_{k=1}^{2}\sum_{j\in Y_k} y_j - \sum_{k=1}^{2}\sum_{l\in\Xi_k^M([2])} z_l\right) \leq 5 \tag{9f}$$

$$\overset{\text{BIP(M)}}{\iff} \quad \sum_{k=1}^{2}\sum_{i\in X_k} x_i + \sum_{k=1}^{2}\sum_{j\in Y_k} y_j - \sum_{k=1}^{2}\sum_{l\in\Xi_k^M([2])} z_l \leq 2. \tag{9g}$$

Inequality (9f) is valid for $BIP(M)$ as it is the sum of five valid inequalities. Further, all variables are binary, which implies that the 2-block inequality (9g) is equivalent to (9f) for the integer points in $BIP(M)$.

For the induction step $n-1 \to n$, we can derive the n-block inequality composed of n blocks M_{X_k,Y_k} for $k \in [n]$ via a combination of the n-many $(n-1)$-block inequalities that can be built out of the blocks $M_{X_{k'},Y_{k'}}$ for $k' \in K_k$, where K_k denotes the index subset of $[n]$ not containing k, i.e., $K_k := [n]\setminus\{k\}$. We can write the n-block inequality as

$$\sum_{i\in[\alpha]} a_i x_i + \sum_{j\in[\beta]} b_j y_j \leq \sum_{l\in[\gamma]} c_l z_l + n, \tag{10}$$

for $a \in \mathbb{R}^\alpha$, $b \in \mathbb{R}^\beta$ and $c \in \mathbb{R}^\gamma$ as defined in (5), (6) and (7), respectively. For $k \in [n]$, denote the corresponding $(n-1)$-block inequality composed of the blocks indexed by K_k as

$$\sum_{i\in[\alpha]} a_i^{K_k} x_i + \sum_{j\in[\beta]} b_j^{K_k} y_j \leq \sum_{l\in[\gamma]} c_l^{K_k} z_l + n - 1. \tag{11}$$

First, we show

$$(1/(n-1)) \sum_{k\in[n]} a_i^{K_k} = a_i$$

for all $i \in [\alpha]$. For any $i \in [\alpha]$ and $k \in [n]$, we have $a_i^{K_k} = a_i - 1$ iff $i \in X_k$, and $a_i^{K_k} = a_i$ otherwise. As we have $|\{k \in [n] : i \in X_k\}| = a_i$,

$$\frac{\sum_{k\in[n]} a_i^{K_k}}{n-1} = \frac{a_i\cdot(a_i-1) + (n-a_i)\cdot a_i}{n-1} = a_i$$

holds. Analogously,

$$(1/(n-1)) \sum_{k\in[n]} b_j^{K_k} = b_j$$

for all $j \in [\beta]$ follows. Next, we show

$$\lfloor (1/(n-1)) \sum_{k\in[n]} c_l^{K_k} \rfloor \leq c_l$$

for all $l \in [\gamma]$. Obviously, removing one block will not cause an element in the matrix to be intersected by more blocks. Therefore, $c_l^{K_k} \leq c_l$ holds for all $l \in [\gamma]$. Moreover, we can neglect the case where c_l is strictly smaller than $n-1$. This is because, for any $p \in \mathbb{N}$, the inequality

$$\frac{n(n-p)}{n-1} \geq n-p+1$$

holds if and only if $p \leq 1$.

Now, for $\lfloor (1/(n-1)) \sum_{k\in[n]} c_l^{K_k} \rfloor$ to be strictly greater than c_l, it would be necessary that $c_l \geq n-1$. This leaves us with two possible cases:

Case 1: $c_l = n$. If there exists some $l' \in [\gamma]$ such that $c_{l'} = n$, then we have

$$\frac{1}{n-1} \sum_{k\in[n]} c_{l'}^{K_k} \leq c_{l'}$$

because $c_{l'}^{K_k} \leq n-1$ for all $k \in [n]$.

Case 2: $c_l = n-1$. Suppose there exists some $\tilde{l} \in [\gamma]$ such that $c_{\tilde{l}} = n-1$. Consider any tuple $(i', j') \in X \times Y$ that belongs to exactly $n-1$ blocks. There is exactly one index $k \in [n]$ for which (i', j') is not contained in $X_k \times Y_k$.

Due to the ordering of the blocks, i.e.,

$$X_{k+1} \subseteq X_k \quad \text{and} \quad Y_k \subseteq Y_{k+1} \quad \text{for all } k \in [n-1],$$

the missing block must be either the first or the last one. Since $n > 2$, this block cannot be the second one.

As a result, we obtain $c_{\tilde{l}}^{K_2} = n-2$. Now, we verify

$$\frac{(n-2)+(n-1)\cdot(n-1)}{n-1} = \frac{n-2}{n-1} + (n-1) < n$$

for $n \geq 3$. Consequently, we conclude that

$$\lfloor (1/(n-1)) \sum_{k\in[n]} c_{\tilde{l}}^{K_k} \rfloor \leq c_{\tilde{l}}.$$

Now summing up all the $(n-1)$-block inequalities (11) for all $k \in [n]$ and dividing result by $n-1$ yields

$$\sum_{i\in[\alpha]} a_i x_i + \sum_{j\in[\beta]} b_j y_j \leq \sum_{l\in[\gamma]} c'_l z_l + n,$$

for some $c' \in \mathbb{R}^\gamma$. Further, $\lfloor c'_l \rfloor \leq c_l$ holds for all $l \in \cup_{k\in K} Z^M(X_k, Y_k)$, as we have already shown. The inequality remains valid when rounding down the coefficients of the z-variables because of the multiple-choice constraints and the z-variables being binary. Now adding the appropriate bound inequalities, namely $0 \leq z_l$ for $l \mid \lfloor c'_l \rfloor < c_l$ we obtain (10). □

The proof demonstrates a hierarchical construction of n-block inequalities from $(n-1)$-block inequalities. Similar to the concept of ranks used for other well-known constraint classes like Chvátal-Gomory cuts [12], Mixed-Integer Rounding inequalities [33], or cuts produced by the Reformulation Linearization Technique [41], we can interpret n-block inequalities in terms of their constructive complexity. This leads to the following definition:

Definition 3 (Rank-m Closure)**.** For a given $m \in \mathbb{N}$, the rank-m closure is defined as the polytope that emerges when adding all n-block inequalities for $n \leq m$ to the bounds and multiple choice constraints:

$$P^{(m)}(M) := \left\{ (x,y,z) \in [0,1]^{\alpha+\beta+\gamma} \mid \text{all } n\text{-block inequalities for } n \leq m,\ \sum_{i=1}^{\alpha} x_i = \sum_{j=1}^{\beta} y_j = \sum_{l=1}^{\gamma} z_l = 1 \right\}.$$

We will use this rank concept for a more structured analysis in our numerical study.

3.2 Characterization of n-Block Inequalities

We begin by addressing the question of when a general inequality of the form

$$\sum_{i\in[\alpha]} a'_i x_i + \sum_{j\in[\beta]} b'_j y_j \leq \sum_{l\in[\gamma]} c'_l z_l + d' \tag{12}$$

with $a', b', c', d' \in \mathbb{N}_0$ constitutes an n-block inequality. If the inequality is already in the canonical form described in Section 3.1, the verification is straightforward: we set $n := d'$ and construct the n blocks $M_{X,Y}$ for $k \in [n]$ by defining $X_k := \{i \in [\alpha] : a'_i \geq k\}$ and $Y_k := \left\{j \in [\beta] : b'_j > n-k\right\}$. This construction guarantees both the nested structure of the blocks (2) and ensures that Conditions (5) and (6) hold. It then suffices to verify Condition (7).

However, this direct approach fails when the inequality is merely equivalent to an n-block inequality rather than identical to one. Indeed, the addition of multiple-choice constraints for the x-, y-, and z-variables, combined with scaling, may produce inequalities that are equivalent to n-block inequalities but for which no subsets X_k and Y_k exist satisfying Conditions (5), (6) and (7) simultaneously. We therefore seek a characterization that is invariant under these operations.

Our approach is to derive necessary and sufficient conditions on the coefficients of (12) that determine whether it can be transformed into an n-block inequality through addition of multiple-choice constraints and scaling. To formulate these conditions, we introduce notation for the extreme coefficient values. Let $\underline{i} \in \operatorname{argmin}_{i\in[\alpha]} a'_i$ and $\bar{i} \in \operatorname{argmax}_{i\in[\alpha]} a'_i$ denote indices of the minimum and maximum coefficients of a', respectively, and define $\underline{j}, \bar{j}$ and $\underline{l}, \bar{l}$ analogously for b' and c'. Since scaling preserves equivalence, we may assume without loss of generality that all coefficients in (12) are integers.

We now derive three conditions that any such equivalent inequality must satisfy. The key observation is that the nested block structure (2) of n-block inequalities imposes strong regularity on the coefficient values. Specifically, since each of the n blocks contains at least one row indexed by $i \in [\alpha]$ and at least one column indexed by $j \in [\beta]$, the maximum coefficients for the x-, y-, and z-variables in any n-block inequality all equal n. To achieve unambiguity we assume the minimum coefficient for z equals zero.

To normalize a general inequality (12) to this form we subtract appropriate multiples of the three multiple-choice constraints:

$$(a'_{\bar{i}} + c'_{\underline{l}} - c'_{\bar{l}}) \cdot \sum_{i\in[\alpha]} x_i = a'_{\bar{i}} + c'_{\underline{l}} - c'_{\bar{l}}, \qquad (b'_{\bar{j}} + c'_{\underline{l}} - c'_{\bar{l}}) \cdot \sum_{j\in[\beta]} y_j = b'_{\bar{j}} + c'_{\underline{l}} - c'_{\bar{l}}, \qquad c'_{\underline{l}} \cdot \sum_{l\in[\gamma]} z_l = c'_{\underline{l}}.$$

For this transformation to yield an inequality where the maximum coefficients of x, y, and z all equal the right-hand side constant d', we require our first condition:

$$a'_{\bar{i}} + b'_{\bar{j}} = c'_{\bar{l}} + d'. \tag{I}$$

Beyond coefficient normalization, the transformation must preserve the fundamental relationship between the coefficients established in Section 3.1. For any n-block inequality, we have

$$c_l = \max_{i,j\in[\alpha]\times[\beta]:M_{ij}=l} \max\{0, a_i + b_j - n\} \quad \forall l \in [\gamma]. \tag{13}$$

Translating this relationship to the transformed coefficients yields our second condition:

$$c'_l = \max_{(i,j)\in[\alpha]\times[\beta]:M_{ij}=l} \max\left\{c'_{\underline{l}}, \left(a'_i - a'_{\bar{i}} + b'_j - b'_{\bar{j}} + c'_{\bar{l}}\right)\right\} \quad \forall l \in [\gamma]. \tag{II}$$

Finally, since n-block inequalities have non-negative coefficients, the transformation must not produce negative values. This requirement imposes our third condition:

$$a'_{\bar{i}} - a'_{\underline{i}} \le c'_{\bar{l}} - c'_{\underline{l}}, \qquad b'_{\bar{j}} - b'_{\underline{j}} \le c'_{\bar{l}} - c'_{\underline{l}}. \tag{III}$$

The following lemma establishes that Conditions (I), (II) and (III) provide a complete characterization of inequalities equivalent to n-block inequalities.

Lemma 4. *An inequality of the form $\sum_{i\in[\alpha]} a'_i x_i + \sum_{j\in[\beta]} b'_j y_j \le \sum_{l\in[\gamma]} c'_l z_l + d'$ is equivalent to an n-block inequality for some $n \in \mathbb{N}$ up to addition of multiple-choice constraints and scaling iff Conditions* (I), (II) *and* (III) *are met.*

Proof. Consider the inequality

$$\sum_{i\in[\alpha]} a'_i x_i + \sum_{j\in[\beta]} b'_j y_j \le \sum_{l\in[\gamma]} c'_l z_l + d', \tag{14}$$

where w.l.o.g. a', b', c', and d' shall be integer. Further, we assume that this inequality fulfills Conditions (I)–(III). We now show that via subtraction of multiple-choice constraints, we can transform this inequality to an n-block inequality of the form

$$\sum_{k=1}^{n} \sum_{i\in\bar{X}_k} x_i + \sum_{k=1}^{n} \sum_{j\in\bar{Y}_k} y_j \le \sum_{k=1}^{n} \sum_{l\in\Xi_k^M([n])} z_l + n, \tag{15}$$

by determining the appropriate $n \in \mathbb{N}$ as well as the sets $\bar{X}_k$ and $\bar{Y}_k$, that need to be sorted by inclusion as follows: $\bar{X}_{k+1} \subseteq \bar{X}_k$ and $\bar{Y}_k \subseteq \bar{Y}_{k+1}$ for all $k \in [n-1]$. Additionally $|\cup_{k\in[n]} Z^M(\bar{X}_k, \bar{Y}_k)| < \gamma$ has to hold. By Condition (I), the following sum of multiple-choice constraints is valid for $BIP(M)$:

$$(-a'_{\bar{i}} + c'_{\bar{l}} - c'_{\underline{l}}) \underbrace{\sum_{i\in[\alpha]} x_i}_{=1} + (-b'_{\bar{j}} + c'_{\bar{l}} - c'_{\underline{l}}) \underbrace{\sum_{j\in[\beta]} y_j}_{=1} = \underbrace{-d'}_{=-a'_{\bar{i}}-b'_{\bar{j}}+c'_{\bar{l}}} - c'_{\underline{l}} \underbrace{\sum_{l\in[\gamma]} z_l}_{=1} + c'_{\bar{l}} - c'_{\underline{l}}.$$

Adding this equation to Inequality (14) yields

$$\sum_{i\in[\alpha]} (a'_i - a'_{\bar{i}} + c'_{\bar{l}} - c'_{\underline{l}}) x_i + \sum_{j\in[\beta]} (b'_j - b'_{\bar{j}} + c'_{\bar{l}} - c'_{\underline{l}}) y_j \le \sum_{l\in[\gamma]} (c'_l - c'_{\underline{l}}) z_l + c'_{\bar{l}} - c'_{\underline{l}}. \tag{16}$$

Note that now the maximum coefficient for each set of variables x, y and z equals $c'_{\bar{l}} - c'_{\underline{l}}$. Additionally, each coefficient is non-negative due to Condition (III). Now define for $k \in [c'_{\bar{l}} - c'_{\underline{l}}]$ the subsets

$$X_k := \{i \in [\alpha] \mid a'_i - a'_{\bar{i}} + c'_{\bar{l}} - c'_{\underline{l}} \ge k\}, \qquad Y_k := \{j \in [\beta] \mid b'_j - b'_{\bar{j}} > -k\}.$$

Each $i \in [\alpha]$ is contained in $(a'_i - a'_{\bar{i}} + c'_{\bar{l}} - c'_{\underline{l}})$-many sets in $\left\{X_k : k \in [c'_{\bar{l}} - c'_{\underline{l}}]\right\}$. Similarly, each $j \in [\beta]$ is contained in $(b'_j - b'_{\bar{j}} + c'_{\bar{l}} - c'_{\underline{l}})$-many sets in $\left\{Y_k : k \in [c'_{\bar{l}} - c'_{\underline{l}}]\right\}$. Further, let $n := c'_{\bar{l}} - c'_{\underline{l}}$. Now, for (16) to

be an n-block inequality it remains to show that each $l \in [\gamma]$ lies in the intersection of $c'_{\bar{l}} - c'_{\underline{l}}$ and not more blocks from $\{X_k \times Y_k : k \in [n]\}$, i.e.,

$$\max\left\{k \in [c'_{\bar{l}} - c'_{\underline{l}}] : l \in \Xi_k^M([c'_{\bar{l}} - c'_{\underline{l}}])\right\} = c'_{\bar{l}} - c'_{\underline{l}} \quad \forall l \in [\gamma].$$

Namely, if for any $l' \in [\gamma]$ we have $l' \in \Xi_k^M([c'_{\bar{l}} - c'_{\underline{l}}])$ for some $k > 1$, then $l' \in \Xi_{k-1}^M([c'_{\bar{l}} - c'_{\underline{l}}])$ follows trivially. The sorting of X_k and Y_k implies that for a pair $(i, j) \in [\alpha] \times [\beta]$ to be in $X_{k'} \times Y_{k'}$ for some $k' \in [c'_{\bar{l}} - c'_{\underline{l}}]$, the conditions $i \in X_k$ for $k \in [k']$ and $j \in Y_k$ for $k \in \left\{k', \ldots, c'_{\bar{l}} - c'_{\underline{l}}\right\}$ have to hold. In particular, for the defined sets $\bar{X}_k$ and $\bar{Y}_k$ for $k \in [c'_{\bar{l}} - c'_{\underline{l}}]$, the number of blocks containing the cell (i, j) of M can be calculated as

$$|\left\{k \in [c'_{\bar{l}} - c'_{\underline{l}}] : (i, j) \in X_k \times Y_k\right\}| = \max\left\{0, \left(a'_i - a'_{\bar{i}} + c'_{\bar{l}} - c'_{\underline{l}} + b'_j - b'_{\bar{j}} + c'_{\bar{l}} - c'_{\underline{l}} - n\right)\right\}.$$

Therefore, we need to have

$$\max_{(i,j)\in[\alpha]\times[\beta]:M_{ij}=l} \max\left\{0, \left(a'_i - a'_{\bar{i}} + c'_{\bar{l}} - c'_{\underline{l}} + b'_j - b'_{\bar{j}} + c'_{\bar{l}} - c'_{\underline{l}} - n\right)\right\} = c'_{\bar{l}} - c'_{\underline{l}}.$$

This is indeed equivalent to Condition (II). Thus, we have shown that any inequality of the form $\sum_{i\in[\alpha]} a'_i x_i + \sum_{j\in[\beta]} b'_j y_j \leq \sum_{l\in[\gamma]} c'_l z_l + d'$ is equivalent to an n-block inequality for some $n \in \mathbb{N}$ if Conditions (I),(II) and (III) are met.

The reverse implication, i.e., every inequality equal to an n-block inequality for $n \in \mathbb{N}$ up to addition of multiple-choice constraints and scaling fulfills Conditions (I),(II) and (III), follows directly from their derivation. □

4 Facets

Facets are the tightest possible linear cuts which can be added to the description of $BIP(M)$ and are therefore useful for the branch-and-cut algorithm for solving optimization problems over $BIP(M)$. In the following it is shown that the so-far described classes of valid inequalities namely n-block and bound inequalities are sufficient to fully describe $BIP(M)$. Additionally, we introduce a separation algorithm and a preprocessing routine to efficiently make use of these inequalities in a branch-and-cut procedure.

4.1 Convex Hull

Lemma 5. *All facets of $BIP(M)$ are induced by either n-block inequalities or lower bounds.*

Proof. Let F be a facet of $BIP(M)$ which is induced by the valid inequality

$$\sum_{i\in[\alpha]} a_i x_i + \sum_{j\in[\beta]} b_j y_j \leq \sum_{l\in[\gamma]} c_l z_l + d, \tag{17}$$

$a', b', c', d' \in \mathbb{N}_0$. Further, let $V = \left\{v_{t_1}, \ldots, v_{t_v}\right\}$ be a set of affine independent vertices for

$$v := \dim(BIP(M))$$

with $V \subseteq F$. By Lemma 1, all vertices in V have the form

$$v_{t_k} = e_{t_k^x} + e_{\alpha + t_k^y} + e_{\alpha+\beta+M_{t_k^x t_k^y}}$$

for some $t_k = (t_k^x, t_k^y) \in [\alpha] \times [\beta]$. The tuple t_k sufficiently characterizes the vertex v_{t_k}. It indicates that $x_i = 1$ for $i = t_k^x$, $x_i = 0$ otherwise and $y_j = 1$ for $j = t_k^y$, $y_j = 0$ otherwise. If there is an index $i' \in [\alpha]$ such that there is no vertex in V fulfilling $x_{i'} = 1$, then F lies on the hyperplane

$\{(x,y,z) \in \mathbb{R}^{\alpha+\beta+\gamma} : x_{i'} = 0\}$ and since we can rule out that this hyperplane is a superset of $BIP(M)$, F is induced by the bound inequality $x_i \geq 0$. This holds analogously for $j \in [\beta]$ and $l \in [\gamma]$. Note that $BIP(M) \not\subset \{(x,y,z) \in \mathbb{R}^{\alpha+\beta+\gamma} : z_l = 0\}$ for all $l \in [\gamma]$ follows from the assumption that each $l \in [\gamma]$ is contained in M.

Now assume that for all $i \in [\alpha]$ there is at least one $k' \in [\nu]$ with $t^x_{k'} = i$, and that the same holds for all $j \in [\beta]$ and $l \in [\gamma]$. W.l.o.g., we can assume

$$a_{t^x_\kappa} = b_{t^y_\kappa} = c_{M_{t^x_\kappa t^y_\kappa}} = d = 0$$

for one $\kappa \in [\nu]$ where $M_{t^x_\kappa t^y_\kappa} = \underline{l}$ since any inequality can be transformed to this form by subtracting multiple-choice constraints. Inserting those informations in (17) implies that all vertices v_{t_k} in V fulfill the equation $a_{t^x_k} + b_{t^y_k} = c_{M_{t^x_k t^y_k}}$. We now want to show that Conditions (I), (II) and (III) from Lemma 4 hold.

First, we verify

$$a_{\bar{i}} + b_{\bar{j}} = c_{\bar{l}}.$$

By assumption, there is a $k' \in [\nu]$ for which $M_{t^x_{k'} t^y_{k'}} = \bar{l}$ holds, hence $a_{t^x_{k'}} + b_{t^y_{k'}} = c_{\bar{l}}$. Now consider the vertex characterized by the tuple $(\bar{i}, \bar{j})$. Since the inequality defining F must be valid for this vertex, we have $a_{\bar{i}} + b_{\bar{j}} \leq c_{M_{\bar{i}\bar{j}}} \leq c_{\bar{l}}$ and therefore $a_{t^x_{k'}} = a_{\bar{i}}$ and $b_{t^y_{k'}} = b_{\bar{j}}$. This implies $a_{\bar{i}} + b_{\bar{j}} = c_{\bar{l}}$, which certifies Condition (I).

Now, we show

$$c_l = \max_{i,j \in [\alpha]\times[\beta] : M_{ij} = l} \max\{0, a_i + b_j\} \tag{18}$$

for all $l \in [\gamma]$. For all $k' \in [\nu]$ for which $M_{t^x_{k'} t^y_{k'}} = l$ holds, we have $a_{t^x_{k'}} + b_{t^y_{k'}} = c_l$. Thus, there are $i \in [\alpha]$ and $j \in [\beta]$ such that $a_i + b_j = c_l$ holds. The validity of the considered inequality for $BIP(M)$ implies $a_i + b_j \leq c_{M_{ij}}$ for all $i, j \in [\alpha] \times [\beta]$. This validates Condition (II).

Finally, we have to show

$$a_{\underline{i}} \geq a_{\bar{i}} - c_{\bar{l}}, \qquad b_{\underline{j}} \geq b_{\bar{j}} - c_{\bar{l}}.$$

To this end, define the two subsets $X := \{i \in [\alpha] \setminus \{\bar{i}\} : a_i < 0\}$ and $Y := \{j \in \beta \setminus \{\bar{j}\} : b_j < 0\}$ and suppose that X or Y is non-empty. Lifting these selected coefficients leads to a valid block inequality dominating (17), contradicting the assumption that (17) is facet-defining. Consider the inequality

$$\sum_{i \in [\alpha]} a''_i x_i + \sum_{j \in [\beta]} b''_j y_j \leq \sum_{l \in [\gamma]} c_l z_l + d', \tag{19}$$

where $a''_i = 0$ holds for all $i \in X$, and $a''_i = a'_i$ otherwise, and where $b''_j = 0$ holds for all $j \in Y$, and $b''_j = b'_j$ otherwise. We can construct the sets

$$\bar{X}_k := \{i \in [\alpha] : a''_i \geq k\}$$

and

$$\bar{Y}_k := \left\{j \in [\beta] : b''_j > d' - k\right\}$$

and set $n' := c_{\bar{l}}$. Now as in the proof of Lemma 4 the number of blocks containing the cell (i,j) of M can be calculated as

$$|\{k \in [d'] : (i,j) \in \bar{X}_k \times \bar{Y}_k\}| = \max\left\{0, a''_i + b''_j - n'\right\}.$$

Together with (18) for the transformed variables,

$$c_l = \max_{i,j \in [\alpha]\times[\beta] : M_{ij} = l} \max\{0, a_i + b_j\},$$

we observe the equivalence of (19) and the following n'-block inequality, which is valid for $BIP(M)$:

$$\sum_{k=1}^{\bar{n}} \sum_{i \in \bar{X}_k} x_i + \sum_{k=1}^{\bar{n}} \sum_{j \in \bar{Y}_k} y_j - \sum_{k=1}^{\bar{n}} \sum_{l \in \Xi_k^M([\bar{n}])} z_l \leq n'.$$

Since Inequality (19) dominates Inequality (17), the latter cannot be facet-defining, which contradicts the assumption. Thus, Condition (III) holds as well. Altogether, this means that (17) is equivalent to an n-block inequality. □

Theorem 6. *The full convex-hull description of $BIP(M)$ is given by the rank-$\bar{n}$ closure for some fixed $\bar{n} \in \mathbb{N}$.*

4.2 Separating n-Block Inequalities

To support a branch-and-cut algorithm by adding useful cuts we develop a separation routine which identifies n-block inequalities which cut off a given non-integer point with maximum violation. As shown in Section 3.2 there are many different inequalities equivalent up to addition of multiple-choice constraints and scaling. Hence, we need to find a unique representation for these cuts.

Definition 7. An inequality of the form $\sum_{i\in[\alpha]} a_i x_i + \sum_{j\in[\beta]} b_j y_j \leq \sum_{l\in[\gamma]} c_l z_l + d$ is called a *normalized block inequality* if it is equivalent to an n-block inequality and if $\min_{l\in[\gamma]} c_l = 0$ as well as $\max_{i\in[\alpha]} a_i = \max_{j\in[\beta]} b_j = \max_{l\in[\gamma]} c_l = 1$ hold.

Note that any n-block inequality can be transformed to a normalized n-block inequality by subtracting multiples of the multiple-choice constraints until $\min_{l\in[\gamma]} c_l = 0$ as well as $\max_{i\in[\alpha]} a_i = \max_{j\in[\beta]} b_j = \max_{l\in[\gamma]} c_l = d$ hold and then dividing by $\max_{l\in[\gamma]} c_l$, which also leads to $d = 1$. This is possible because of Condition (I) from Lemma 4. As a consequence, all normalized block inequalities also satisfy this condition. We can make use of the fact that for normalized block inequalities, Condition (II) simplifies to

$$c_l = \max_{i,j\in[\alpha]\times[\beta]: M_{ij}=l} \max\{0, a_i + b_j - 1\} \quad \forall l \in [\gamma].$$

Further, Condition (III) can be ensured by bounding the a- and b-variables from below by zero. This allows us to state an optimization problem to find normalized block inequalities which are maximally violated by a given not necessarily integer point $q = (\bar{x}, \bar{y}, \bar{z}) \notin BIP(M)$ with $q \geq 0$:

$$\max_{(a,b,c)\in P^{SEP}(M)} \quad \sum_{i\in[\alpha]} a_i \bar{x}_i + \sum_{j\in[\beta]} b_j \bar{y}_j - \sum_{l\in[\gamma]} c_l \bar{z}_l - 1, \tag{20}$$

$$P^{SEP}(M) := \left\{ a \in [0,1]^\alpha, b \in [0,1]^\beta, c \in [0,1]^\gamma : c_l \geq a_i + b_j - 1 \quad \forall l \in [\gamma],\ (i,j) \in [\alpha]\times[\beta] : M_{ij} = l \right\}$$

The variables $(a, b, c) \in [0,1]^{\alpha+\beta+\gamma}$ are the left-hand side coefficients of the normalized block inequality we search for while the constraints enforce Conditions (I) - (III).

Theorem 8. *Assume $q \geq 0$ and $\sum_{i\in[\alpha]} \bar{x}_i = \sum_{j\in[\beta]} \bar{y}_j = \sum_{l\in[\gamma]} \bar{z}_l = 1$. Then every vertex of $P^{SEP}(M)$ which is optimal for* (20) *yields the coefficients of a normalized block inequality.*

Proof. Let $s = (\tilde{a}, \tilde{b}, \tilde{c})$ a vertex of $P^{SEP}(M)$. We show that conditions (I) - (III) are satisfied. Since s is a vertex of $P^{SEP}(M)$ and optimal for (20), there is no $s' \in P^{SEP}(M)$, $s' \neq s$ which has the same or a higher objective value as s.

We have to verify that the highest value in each variable set is equal to one, $\tilde{a}_{\bar{i}} = \tilde{b}_{\bar{j}} = \tilde{c}_{\bar{l}} = 1$.
If none of the highest values is equal to one, we can multiply all values by some positive factor staying feasible and increasing the objective value. We know $\tilde{a}_{\bar{i}} + \tilde{b}_{\bar{j}} - 1 = \tilde{c}_{\bar{l}}$, otherwise we could decrease $\tilde{c}_{\bar{l}}$ while again staying feasible and increasing the objective value. Therefore, *w.l.o.g.* assume $\tilde{a}_{\bar{i}} = 1$. If

now $\tilde{b}_{\bar{j}} \neq 1$ it follows $\tilde{c}_{\bar{l}} \neq 1$. Now we add $1 - \tilde{b}_{\bar{j}}$ to all values of $\tilde{b}$ and $\tilde{c}$. The multiple-choice constraints lead to the fact that we arrive at a feasible point s' which has the same objective function value as s. This proves $\tilde{a}_{\bar{i}} = \tilde{b}_{\bar{j}} = \tilde{c}_{\bar{l}} = 1$, conditions (I) and (III) follow trivially.

The non-bound constraints in $P^{SEP}(M)$ directly imply

$$\tilde{c}_l \geq \max_{i,j \in [\alpha]\times[\beta]: M_{ij}=l} \max\{0, \tilde{a}_i + \tilde{b}_j - 1\} \quad \forall l \in [\gamma].$$

The equality and therefore condition (II) is obvious given that increasing values of $\tilde{c}$ leads to decreasing the objective value. □

Remark 1. Normalized block inequalities can be separated from points satisfying multiple-choice constraints in polynomial time.

To separate only 1-block inequalities to obtain the rank-1 closure it suffices to limit the solution space to binary values of a, b and c. By adding the constraint $\sum_{l\in[\gamma]} c_l = \nu$ the number of z-variables in the resulting inequality is restricted to a chosen value ν.

There are cases in which a non-facet n-block inequality is more violated by an infeasible point than any facet. The following is an example for this exception.

Example 2. Consider the relation matrix

$$M = \begin{pmatrix} 2 & 5 & 1 \\ 2 & 1 & 4 \\ 3 & 4 & 3 \end{pmatrix}$$

and $p = (0, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 0, \frac{1}{2}, 0, 0, 0, 0, 1)$. The constraint

$$\frac{1}{2}x_1 + x_2 + x_3 + y_1 + \frac{1}{2}y_2 + y_3 \leq \frac{1}{2}z_1 + z_2 + z_3 + z_4 + 1$$

is violated by 1. It can be conically combined by the facets

$$\begin{aligned} (0.5\cdot) \quad & x_1 + x_2 + y_1 + y_2 + y_3 \leq z_1 + z_2 + z_4 + z_5 + 1 \\ (1.5\cdot) \quad & \frac{1}{3}x_1 + \frac{2}{3}x_2 + x_3 + y_1 + \frac{2}{3}y_2 + y_3 \leq \frac{1}{3}z_1 + \frac{2}{3}z_2 + z_3 + \frac{2}{3}z_4 + 1 \end{aligned}$$

and the multiple-choice constraints

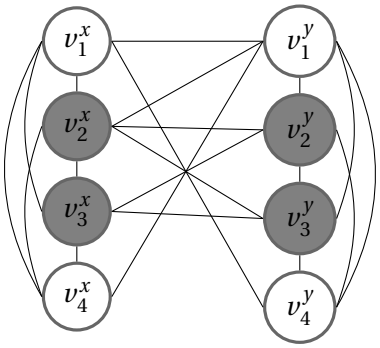
$$\begin{aligned} (0.5\cdot) \quad & -x_1 - x_2 - x_3 \leq -1 \\ (1\cdot) \quad & -y_1 - y_2 - y_3 \leq -1 \\ (0.5\cdot) \quad & z_1 + z_2 + z_3 + z_4 + z_5 \leq 1 \end{aligned}$$

and is therefore not itself a facet. But it nevertheless is more violated by p than the facets it can be assembled from and in fact any facet of $BIP(M)$.

4.3 Precomputing 1-Block Inequalities Using Cliques

Experience shows that 1-block inequalities form the largest part of the facets of $BIP(M)$. Since it is relatively computationally easy to find good 1-block inequalities for $BIP(M)$ it can be useful to add some of them before starting the optimization process. The problem to find a block in M as large as possible which contains only a given subset Z of $[\gamma]$ can be formulated as a clique problem with a quadratic objective function. For that, we build a graph $G^C(M) = (V^C(M), E^C(M))$ whose nodes $V^C(M) = V_X^C(M) \cup V_Y^C(M) = \{v_1^x, ..., v_\alpha^x\} \cup \{v_1^y, ..., v_\beta^y\}$ correspond to either a row or a column of M.

Now, edges are introduced such that the subgraphs of $G^C(M)$ induced by the variable set $V_X^C(M)$ and $V_Y^C(M)$, respectively, are complete. Additionally, two nodes $v_i^x \in V_X^C(M)$ for $i \in [\alpha]$ and $v_j^y \in V_Y^C(M)$ for $j \in [\beta]$ are connected by an edge if the z-index M_{ij} is contained in Z. The selected nodes in a clique in $G^C(M)$ correspond to the rows and columns of M forming a block which only contains indices in the given subset Z of $[\gamma]$. If all rows and all columns of a block A are contained in a block B and both A and B contain the same set of z-indices, the inequality induced by A is dominated by the inequality induced by B. Hence, to make the block as big as possible, we want to optimize over its volume. The quadratic objective function is given as the number of selected nodes in $V_X^C(M)$ times the number of selected nodes in $V_Y^C(M)$. We can either solve this clique problem exactly or use a heuristic.



2	4	4	2
2	2	1	4
4	1	1	4
1	3	4	3

Figure 4: Clique problem to find maximum 1-block only containing indices in $\{1,2\}$.

Example 1 (Example 1 continued)**.** Figure 4 shows a Matrix M and the corresponding graph $G^C(M)$. To find the largest 1-block in M which contains only the z-indices 1 and 2, we find a clique in $G^C(M)$ which maximizes the function $f(\chi) = \left(\sum_{i=1}^4 \chi_{v_i^x}\right) \cdot \left(\sum_{j=1}^4 \chi_{v_j^y}\right)$, where the binary variable χ indicates the selection of a node. The optimal solution forms the grey shaded block of size 4 in Figure 4. The 1-block inequality $x_2 + x_3 + y_2 + y_3 \leq z_1 + z_2 + 1$ can be added to the model.

5 Chained Bipartite Implication Polytopes

Inspired by an application in stochastic programming, we now chain several bipartite implication related variable sets with multiple-choice constraints over a given planning horizon $t \in [T]$, $T \in \mathbb{N}$. For each time step $t \in [T]$, there are two vectors of binary variables with multiple-choice condition $x^t \in \{0,1\}^{\alpha_t}$ and $y^t \in \{0,1\}^{\beta_t}$ and a matrix M^t describing the implications between the variable sets, which are given as follows:

$$\text{IF } x_i^t = 1 \text{ for some } i \in [\alpha_t] \text{ AND } y_j^{t-1} = 1 \text{ for } j \in [\beta_{t-1}], \text{ THEN } y_{j'}^t = 1,$$

for all $t \in [T]$, where $j' := M_{ij}^t$ and $y^0 \in \{0,1\}^{\beta_0}$ is a given vector with multiple-choice condition. The set of feasible points can thus be expressed as all binary vectors $(x^t, y^t) \in \{0,1\}^{\alpha_t+\beta_t+\gamma_t}$ for $t \in [T]$ which fulfill the following constraints:

$$\sum_{i \in [\alpha_t]} x_i^t = \sum_{j' \in [\beta_t]} y_{j'}^t = 1 \qquad \forall\, t \in [T] \tag{22}$$

$$x_i^t + y_j^{t-1} \leq y_{M_{ij}^t}^t + 1 \qquad \forall\, t \in [T], \forall\, i \in [\alpha_t], \forall\, j \in [\beta_{t-1}]. \tag{23}$$

Let $\mathcal{M} := \{M^t : t \in [T]\}$ denote the set of all implication matrices used in the instance. By

$$S(\mathcal{M}) := \left\{(x^1, \ldots, x^T, y^1, \ldots, y^T) \in \{0,1\}^{\sum_{t\in[T]}(\alpha_t+\beta_t)} \quad \forall\, t \in [T] : (22), (23)\right\},$$

we denote the binary feasible points for Constraints (22) and (23). We then call the convex hull of these feasible points $BIP(\mathcal{M}) := \text{conv}(S(\mathcal{M}))$.

Lemma 9. *There are $\prod_{t\in T}\alpha_t$ vertices of $BIP(\mathcal{M})$.*

Proof. Each point in $S(\mathcal{M})$ can be identified by the x-variables which are set to one. For a given vector $y^0 \in \{0,1\}^{\beta_0}$, the values of the variables $y^t_{j'}$ can be derived recursively via $y^t_{j'} = \sum_{(i,j):M^{t-1}_{ij}=j'} x^t_i y^{t-1}_j$. Further, each of the $\prod_{t\in T}\alpha_t$ configurations of possible values for the x-variables lead to feasible points in $S(\mathcal{M})$. As $BIP(\mathcal{M})$ is the convex hull of a set of binary points, these points are all vertices of $BIP(\mathcal{M})$. □

To derive a full outer description of $BIP(\mathcal{M})$ we model it as an instance of the *clique problem with multiple-choice constraints (CPMC) under a cycle-free dependency graph* which has been studied in [7]. In CPMC the task is to find an m-clique in an m-partite graph $G = (V,E)$. This can be seen as a clique problem with additional multiple-choice constraints on the selection of the nodes from each subset in the m-partition $\mathcal{V}$ of V. The convex hull polytope for an instance $(G,\mathcal{V})$ is denoted as $P^{\text{CPMC}}(G,\mathcal{V})$.

We first construct an undirected graph $G^{\mathcal{M}} = (V^{\mathcal{M}}, E^{\mathcal{M}})$ as follows. For all $t \in [T]$, each variable x^t_i, $i \in [\alpha_t]$ and $y^t_{j'}$, $j' \in [\beta_t]$, is represented by a node $v_{x^t_i}$ or $v_{y^t_{j'}}$ in $V^{\mathcal{M}}$, respectively. For each cell ij' in the implication matrices M^t, $t \in [T]$, we further introduce a node $v_{m^t_{ij'}}$. Each node is assigned to exactly one node subset, namely $v_{x^t_i}$ to V_{x^t}, $v_{y^t_{j'}}$ to V_{y^t} and $v_{m^t_{ij}}$ to V_{m^t}, $t \in [T]$, $i \in [\alpha_t]$, $j' \in [\beta_t]$, $j \in [\beta_{t-1}]$. Figure 5 depicts this procedure for two chained bipartite implication instances. Additionally, we introduce the node subset V_{y^0} containing only one node $v_{y^0_j}$, where $y^0_j = 1$. These node subsets constitute a partition $\mathcal{V}^{\mathcal{M}}$ of $V^{\mathcal{M}}$ into disjoint stable subsets. Now we introduce edges such that for each $t \in [T]$ the subgraph of G induced by all nodes in V_{x^t}, $V_{y^{t-1}}$ and V_{y^t} is a complete tripartite graph on the three variable sets. Additionally, each node $v_{m^t_{ij}}$, $i \in [\alpha_t]$, $j \in [\beta_{t-1}]$, is connected to the nodes $v_{x^t_i}$, $v_{y^{t-1}_j}$ and $v_{y^t_{j'}}$, where j' is the entry M^t_{ij} in the corresponding implication matrix. We can now decompose $G^{\mathcal{M}}$ into subgraphs $G^{\mathcal{M}}_1, \ldots, G^{\mathcal{M}}_T$, where $G^{\mathcal{M}}_t = (V^{\mathcal{M}}_t, E^{\mathcal{M}}_t)$ is induced by the node set $V^{\mathcal{M}}_t := V_{y^{t-1}} \cup V_{x^t} \cup V_{y^t} \cup V_{m^t}$ for all $t \in [T]$ and connect each pair of nodes which are not in the same subgraph.

Observation 3. An integer point in $BIP(\mathcal{M})$ corresponds to an integer point in $P^{\text{CPMC}}(G^{\mathcal{M}}, \mathcal{V}^{\mathcal{M}})$.

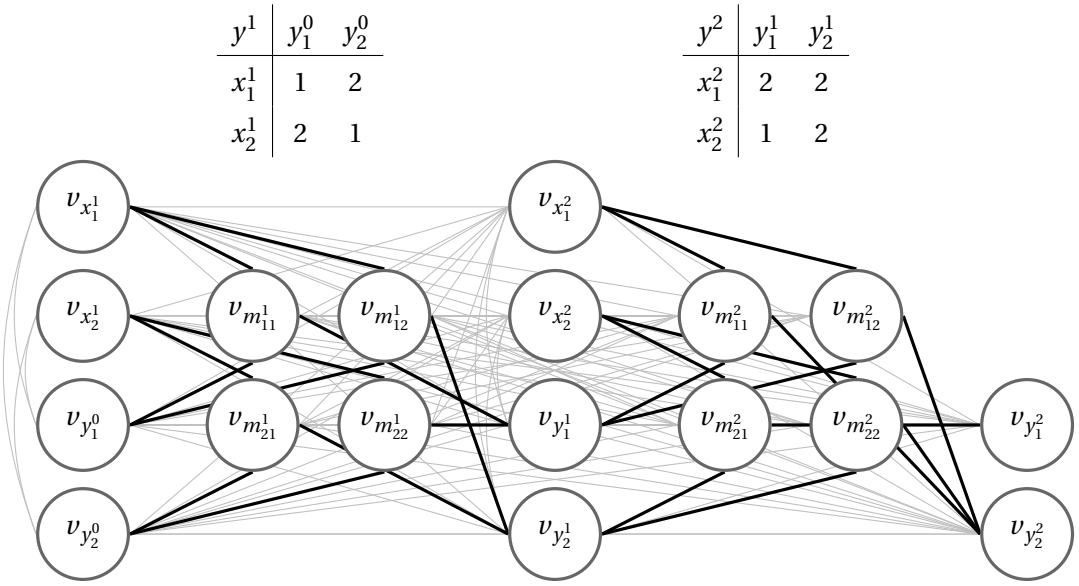


Figure 5: Exemplary construction of the compatibility graph for two chained bipartite implication instances.

The *dependency graph* $\mathcal{G} = (\mathcal{V},\mathcal{E})$ of a CPMC instance $(G,\mathcal{V})$ is defined as follows. Each node partition set in G is represented by a node in $\mathcal{G}$. Two nodes V_i and V_j are connected by an edge if and

only if there exist two nodes $v \in V_i$ and $w \in V_j$ such that there is no edge connecting v and w in G. The dependency graph for the CPMC instance constructed above is depicted in Figure 6.

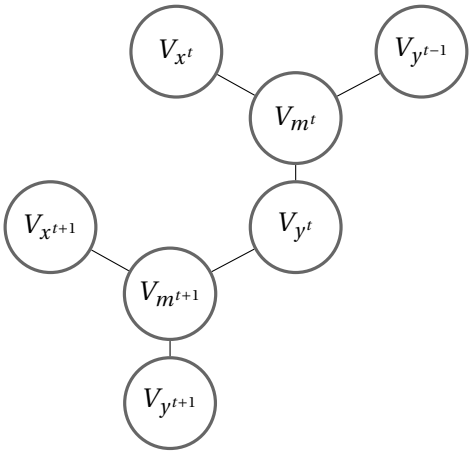


Figure 6: Dependency graph for the CPMC extension of $BIP(\mathscr{M})$.

It can be observed that $\mathscr{G}$ is a forest, which is the prerequisite for the main result of [7] giving a complete description for $P^{\text{CPMC}}(G,\mathcal{V})$.

Theorem 10 ([7], Theorem 3.1)**.** *Let $\mathscr{I} = (G,\mathcal{V})$ be an instance of (CPMC) with a cycle-free dependency graph. Then $P^{CPMC}(G,\mathcal{V})$ is completely described by the constraints*

$$\sum_{v\in U} x_v = 1 \quad \forall U \in \mathcal{V} \tag{24a}$$

$$\sum_{v\in S} x_v \leq 1 \quad \forall \text{ stable sets } S \subseteq V \tag{24b}$$

$$x_v \geq 0 \quad \forall v \in V. \tag{24c}$$

Theorem 10 implies that the convex hull of the feasible points in the extended formulation of the chained BIP is given by the multiple-choice constraints (24a) on the variable sets V_{x^t}, V_{y^t} and V_{m^t} for all $t \in [T]$, the stable-set constraints (24b) and the non-negativity constraints (24c) for all variables.

Note that the nodes in the intersection of two of the subgraphs $G_1^{\mathscr{M}},\dots,G_T^{\mathscr{M}}$ form a stable set in $G^{\mathscr{M}}$. Since the stable-set polytope for $G^{\mathscr{M}}$ is identical to the clique polytope for its complement graph $\bar{G}^{\mathscr{M}}$, we can use the following result from [13] to state that the outer description for $P^{\text{CPMC}}(G^{\mathscr{M}},\mathcal{V}^{\mathscr{M}})$ decomposes into the outer descriptions for each of the polytopes $P^{\text{CPMC}}(G^{\{M^t\}},\mathcal{V}^{\{M^t\}})$ for all $t \in [T]$.

Theorem 11 ([13], Theorem 4.1)**.** *Let $G^1 = (V^1,E^1)$ and $G^2 = (V^2,E^2)$ be graphs such that $G^1 \cap G^2 := (V^1 \cap V^2, E^1 \cap E^2)$ is complete and let $A_1x_1 \leq b_1$, $A_2x_2 \leq b_2$ be complete descriptions of the stable-set polytopes of G^1 and G^2, respectively. Then the union of these linear systems is a complete description of the stable-set polytope of the graph $G^1 \cup G^2 := (V^1 \cup V^2, E^1 \cup E^2)$.*

To obtain an outer description of $BIP(\mathscr{M})$, we use Fourier-Motzkin elimination to project the variables m_{ij}^t, $i \in [\alpha_t]$, $j \in [\beta_{t-1}]$, $t \in [T]$, out of the linear system describing the convex hull of $P^{\text{CPMC}}(G^{\mathscr{M}},\mathcal{V}^{\mathscr{M}})$. Each variable m_{ij}^t, $i \in [\alpha_t]$, $j \in [\beta_{t-1}]$, $t \in [T]$ is included in the inequality system describing the convex hull of $P^{\text{CPMC}}(G^{\{M^t\}},\mathcal{V}^{\{M^t\}})$ for exactly one $t \in [T]$. Therefore, the Fourier-Motzkin elimination can be performed for each $t \in [T]$ separately. This implies that the linear system describing the convex hull of $BIP(\mathscr{M})$ decomposes into the inequalities describing the convex hull of $BIP(M)$ for all $M \in \mathscr{M}$.

Corollary 12. *The polytope $BIP(\mathscr{M})$ is completely described by the non-negativity constraints and all n-block inequalities which are facet-defining for any of the polytopes $BIP(M)$, $M \in \mathscr{M}$.*

6 Computational Results

We now conduct numerical experiments to validate our theoretical findings and evaluate the practical impact of n-block inequalities. Our computational study follows a three-part strategy:

- First, we analyze the effectiveness of our separation routine and clique-based preprocessing algorithm on random instances of individual BIPs. This validates the core strength of our approach on the fundamental polytope structure.
- Second, we examine the performance on tree-like chained BIP structures through a real-world application in fixed recourse stochastic programming, specifically railway timetabling. Since our theoretical results guarantee complete convex hull descriptions for tree-like chainings, this study demonstrates the practical value of this special case.
- Finally, we investigate the broader applicability of our cutting planes on problems that involve cyclic BIP chainings, using instances from the quadratic assignment problem (QAP) benchmark library. Our results highlight that the proposed cuts substantially improve the formulation of QAP instances that involve general chaining structures. However, we emphasize that our goal is not to compete directly with specialized QAP solvers or the most efficient QAP formulations, but rather to illustrate the strength and general usefulness of our cuts for problems exhibiting chaining characteristics.

All algorithms were implemented in Python 3.10.13 using Gurobi 11.0.0 with default parameter settings to solve mixed-integer problems. We performed the calculations on a server with an Intel Xeon E3-1240 v6 CPU, 32 GB RAM, 4 cores, HT disabled and 3.70 GHz base frequency.

6.1 Random Matrix Tests

In this first study, we focus on validating our core algorithmic contributions by analyzing individual BIP instances without chaining.

To estimate the benefit of adding block inequalities to problems which lack observable structure in the relations between the three variable sets indexed in $[\alpha]$, $[\beta]$, and $[\gamma]$, we first conduct performance tests with random relation matrices. To this end, we insert n-block inequalities into the problem at two access points: before the solution algorithm is started and during the branch-and-cut procedure. First, we focus on the rank-1 closure, which involves adding all 1-block inequalities to the initial formulation.

6.1.1 Precomputing 1-Block Inequalities Using Cliques Tests

We evaluate the impact of 1-block inequalities generated by the clique-based algorithm in Section 4.3 on quadratic matrices of various sizes with different ratios for the number of z-indices in relation to the matrix size. For each configuration of α, β, and γ, we perform 300 runs on randomly generated relation matrices to stabilize the results. Instances where the LP relaxation solution is integer directly are discarded. Each run involves optimizing a cost function over $BIP(M)$. We select random cost coefficients for the x- and y-variables and determine the cost coefficients for the z-variables such that the mean cost of all integer points in $BIP(M)$ equals zero. This is achieved by iterating over all feasible integer points of $BIP(M)$. By normalizing the mean cost to zero, we aim to objectively assess the performance of the developed methods without bias towards specific elements in $[\gamma]$.

As the set $[\gamma]$ increases in size, the number of different combinations of z-indices also increases. Since 1-block inequalities can be built for each subset of $[\gamma]$, 1-block inequalities for only one $l \in [\gamma]$ make up a relatively small part of the total set of facets of $BIP(M)$, assuming that the facets are evenly distributed across the subsets of z-indices they contain. To evaluate this distribution, we measure the relative reduction of the integrality gap when adding all 1-block inequalities for different-sized subsets of $[\gamma]$ in the corresponding blocks. We define the integrality gap as the difference between the optimal

integer solution value and the optimal value of the linearly relaxed problem. In each run, we alternate between optimizing the LP relaxation of $BIP(M)$ and cutting off the resulting non-integer point using an 1-block inequality containing a fixed number of z-variables found by the adjusted separation algorithm described in Section 4.2. Table 1 shows that increasing the number of combinations of

Config	$\lvert Z^M\rvert \leq 1$	$\lvert Z^M\rvert \leq 2$	$\lvert Z^M\rvert \leq 3$	$\lvert Z^M\rvert \leq 4$	$\lvert Z^M\rvert \leq 5$
$\alpha = \beta = 10, \gamma = 12$	8.27%	71.87%	93.34%	97.59%	98.58%
$\alpha = \beta = 10, \gamma = 20$	2.75%	42.28%	70.22%	86.03%	93.59%
$\alpha = \beta = 10, \gamma = 28$	1.34%	29.69%	52.68%	69.74%	82.31%
$\alpha = \beta = 15, \gamma = 27$	2.25%	40.18%	65.90%	82.88%	92.13%
$\alpha = \beta = 15, \gamma = 45$	0.62%	19.62%	37.59%	53.42%	66.65%
$\alpha = \beta = 15, \gamma = 63$	0.14%	12.50%	24.29%	35.52%	45.91%
$\alpha = \beta = 20, \gamma = 48$	0.55%	24.55%	44.14%	59.79%	73.18%
$\alpha = \beta = 20, \gamma = 80$	0.16%	11.79%	22.27%	31.89%	41.06%
$\alpha = \beta = 20, \gamma = 112$	0.07%	8.26%	16.14%	23.37%	30.26%

Table 1: Integrality gap reduction by the rank-1 closure.

z-variables in the added 1-block inequalities yields solution values of the LP relaxations which are significantly closer to the solution value of the IPs. As a result, more cuts have to be computed, which can slow down the subsequent branch-and-cut process. Therefore the achieved reduction of the integrality gap is relativized by the number of cuts which were produced. In Table 2, the cells of Table 1

Config	$\lvert Z^M\rvert \leq 1$	$\lvert Z^M\rvert \leq 2$	$\lvert Z^M\rvert \leq 3$	$\lvert Z^M\rvert \leq 4$	$\lvert Z^M\rvert \leq 5$
$\alpha = \beta = 10, \gamma = 12$	0.6890%	0.9214%	0.3132%	0.1231%	0.0622%
$\alpha = \beta = 10, \gamma = 20$	0.1373%	0.2014%	0.0520%	0.0139%	0.0043%
$\alpha = \beta = 10, \gamma = 28$	0.0477%	0.0731%	0.0143%	0.0029%	0.0007%
$\alpha = \beta = 15, \gamma = 27$	0.1872%	0.5151%	0.2211%	0.1045%	0.0581%
$\alpha = \beta = 15, \gamma = 45$	0.0309%	0.0934%	0.0278%	0.0086%	0.0031%
$\alpha = \beta = 15, \gamma = 63$	0.0049%	0.0308%	0.0066%	0.0015%	0.0004%
$\alpha = \beta = 20, \gamma = 48$	0.0462%	0.3147%	0.1481%	0.0754%	0.0462%
$\alpha = \beta = 20, \gamma = 80$	0.0082%	0.0561%	0.0165%	0.0051%	0.0019%
$\alpha = \beta = 20, \gamma = 112$	0.0025%	0.0204%	0.0044%	0.0010%	0.0002%

Table 2: Integrality gap reduction by the rank-1 closure per cut.

are divided by the number of possible combinations of z-indices which are contained in the generated blocks $\sum_{k=1}^{|Z^M|} \binom{\gamma}{k}$. For all observed instances, including 1-block inequalities with two z-variables has the biggest impact on the average integrality gap reduction per cut. Building on that finding, we configure the performance test for the clique algorithm to precompute 1-block inequalities such that for each z-index ($Z^M = \{l\}$) and for each pair of z-indices ($Z^M = \{l_1, l_2\}$) we calculate the largest block (X, Y) in M which contains only $l \in Z^M$. We then add the corresponding 1-block cut

$$\sum_{i \in X} x_i + \sum_{j \in Y} y_j \leq \sum_{l \in Z^M} z_l + 1,$$

to the description of $BIP(M)$. We present the achieved integrality gap reductions in Table 3. For small instances $\alpha = \beta = 8$, $\gamma = 4$ the integrality gap is getting closed by almost two thirds. But the amount of gap reduction decreases when increasing the size of M, while keeping its ratio to the number of z-indices constant. Table 4 shows that the average size of the computed blocks $|X| \cdot |Y|$ does not increase for larger matrices M. Therefore, the computed blocks cover a smaller portion of M for larger matrices. Nevertheless, adding 1-block inequalities computed by the presented clique-based

Config	$\gamma/(\alpha\cdot\beta)=0.0625$	$\gamma/(\alpha\cdot\beta)=0.125$	$\gamma/(\alpha\cdot\beta)=0.1875$	$\gamma/(\alpha\cdot\beta)=0.25$
$\alpha=\beta=8$	64.67%	61.55%	46.85%	40.70%
$\alpha=\beta=12$	40.85%	28.65%	22.91%	16.24%
$\alpha=\beta=16$	25.95%	17.19%	14.34%	9.62%
$\alpha=\beta=20$	18.55%	12.05%	9.81%	6.65%

Table 3: Percentage of gap reduction via clique block generation.

Config	$\gamma/(\alpha\cdot\beta)=0.0625$	$\gamma/(\alpha\cdot\beta)=0.125$	$\gamma/(\alpha\cdot\beta)=0.1875$	$\gamma/(\alpha\cdot\beta)=0.25$
$\alpha=\beta=8$	11.67	6.85	4.64	4.43
$\alpha=\beta=12$	9.55	6.17	4.29	4.22
$\alpha=\beta=16$	8.60	5.81	4.11	4.09
$\alpha=\beta=20$	8.02	5.57	3.97	4.00

Table 4: Average size of the maximum blocks.

algorithm to the description of $BIP(M)$ can be beneficial for the solution process if the ratio $\gamma/(\alpha\cdot\beta)$ is small.

6.1.2 Cut Algorithm Tests

Since we established that the class of all n-block inequalities defines the convex hull of $BIP(M)$, we can use a purely n-block-cut based solution algorithm to optimize over $BIP(M)$. This algorithm follows a standard cutting plane approach: we start by solving the LP relaxation of $S(M)$ defined in Section 2, then iteratively separate violated n-block inequalities and add them as cuts until an integer solution is found or optimality is proven. This algorithm is tested in this section. The following test instances were generated in the same way as in the previous section. The presented measurements include the average number of n-block-cuts which were used to separate non-integer solutions (Table 5), what percentage of these cuts are facet-defining (Table 6), and their distribution over the number n of blocks they consist of (Table 7). To check whether added cuts are facet-defining, we can compute the number of affinely independent vertices of $BIP(M)$ that satisfy the inequality with equality since they can be enumerated as stated in Lemma 1.

Config	$\gamma/(\alpha\cdot\beta)=0.04$	$\gamma/(\alpha\cdot\beta)=0.12$	$\gamma/(\alpha\cdot\beta)=0.2$	$\gamma/(\alpha\cdot\beta)=0.28$
$\alpha=\beta=5$	-	1.51	1.97	2.55
$\alpha=\beta=10$	1.80	3.24	5.38	11.54
$\alpha=\beta=15$	2.75	6.07	15.60	26.31
$\alpha=\beta=20$	4.02	10.65	29.97	46.79

Table 5: Number of used cuts.

Table 5 shows that the number of required cuts increases both with the matrix size $\alpha\cdot\beta$ and with the number of z indices in M. The total amount of runtime in the solution process which accounts for the cut generations scales well with the instance size.

The data in Table 6 reveals an interesting pattern: For small matrices with few z-indices (low $\gamma/(\alpha\cdot\beta)$ ratio), almost all separated cuts are facet-defining for the bipartite implication polytope they were generated from. For example, with $\alpha=\beta=10$ and $\gamma/(\alpha\cdot\beta)=0.04$, 99.63% of the cuts are facets. However, as either the matrix size or the ratio of z-indices increases, this percentage drops dramatically. For large instances with many z-indices ($\alpha=\beta=20$ and $\gamma/(\alpha\cdot\beta)=0.28$), only 0.11% of the cuts are facet-defining. As shown in Table 7, 1-block inequalities make up the largest part of the

Config	$\gamma/(\alpha\cdot\beta)=0.04$	$\gamma/(\alpha\cdot\beta)=0.12$	$\gamma/(\alpha\cdot\beta)=0.2$	$\gamma/(\alpha\cdot\beta)=0.28$
$\alpha=\beta=5$	-	98.24%	88.31%	65.80%
$\alpha=\beta=10$	99.63%	91.36%	43.15%	11.82%
$\alpha=\beta=15$	99.88%	70.66%	9.70%	1.25%
$\alpha=\beta=20$	99.75%	40.18%	1.61%	0.11%

Table 6: Facet quota.

Config	$n=1$	$n=2$	$n=3$	$n=4$	$n\geq 5$
$\alpha=\beta=5,\ \gamma=7$	95.05%	4.24%	0.29%	0.43%	0.00%
$\alpha=\beta=10,\ \gamma=28$	96.12%	3.55%	0.17%	0.14%	0.03%
$\alpha=\beta=15,\ \gamma=63$	97.54%	2.23%	0.12%	0.06%	0.06%
$\alpha=\beta=20,\ \gamma=112$	98.12%	1.72%	0.09%	0.06%	0.01%

Table 7: Distribution of n.

used cuts for all tested instances. Hence, even though there is no upper bound on the number of facets for BIPs presented in this paper, the bound on the number $(2^\alpha-1)(2^\beta-1)$ of 1-block inequalities from Observation 2 is numerically a good estimate for the maximum number of facets around an integer solution to the problem.

6.2 Application to Fixed Recourse Stochastic Programming

We now explore a practical application of the BIP within the context of fixed recourse stochastic programming (FRSP), specifically focusing on stochastic railway timetabling. This real-world example involves multiple bipartite implication instances that are interconnected, forming a tree-like chain as detailed in Section 5. The following studies are carried out on a case study for energy-efficient timetable optimization in underground train networks. Note that the model analyzed is, to the best of our knowledge, the most efficient representation of this application case. We employ the fastest MIP solver, Gurobi, with the goal of advancing the state-of-the-art approach to solving stochastic, energy-efficient timetable optimization in underground train networks. The underlying model synchronizes braking and acceleration phases of locally close trains to make use of recuperation energy which braking trains generate. Additionally, power-saving driving behavior is supported. This is done by slightly changing departure times and running times in the train timetable. For every leg in the table, one can choose from a discrete set of departure and running time combinations. The mixed-integer optimization model to minimize the total energy consumption is given by

$$\begin{aligned}
\min\quad & \sum_{t\in T} z_t \\
\text{s.t.}\quad & \sum_{(i,j)\in J,(d,r)\in C_{ij}} p_{ijdrt}\,x_{ijdr} \leq z_t, \quad \forall\, t\in T \\
& z_t \geq 0, \quad \forall\, t\in T \\
& x\in X.
\end{aligned}$$

Finding a feasible timetable $x\in X$ is modeled as a clique problem with multiple-choice constraints. A detailed description of the mathematical model can be found in [7].

The fixed recourse stochastic aspect is present in the scenario extension of the timetabling model. This feature is described in [9] and provides a way to deal with endogenous uncertainties and delays in the operation of the underground network. Decisions for the running- and departure times in the table have influence on the realization of the uncertainties with respect to delays. We now observe the inequalities added for the full recovery model in [9]. The constraints linking the timetable variables

x_{ijdr} and the variables $y_{sij-1d''r''}$ of scenario s for each leg (i,j) and the leg before $(i,j-1)$ with departure times d, d'' and running times r, r'' are given by

$$x_{ijdr} + y_{sij-1d''r''} - 1 \leq y_{sijd'r'}.$$

The departure time d' and running time r' can be calculated from d, d'' and r, r'' as follows:

$$d' = \max\{d, d'' + r'' + \underline{h}_{ij-1}\} + \delta_{sij},$$
$$r' = \max\{\underline{r}_{ij}, r - (d' - d - \delta_{sij}) + \rho_{sij}\}.$$

Here, $\underline{h}_{ij}$ is the minimum dwell time for leg (i,j), $\underline{r}_{ij}$ is the minimum running time for leg (i,j), δ_{sij} is the deviation from the nominal dwell time before leg (i,j) in scenario s, and ρ_{sij} denotes the deviation from the nominal running time for leg (i,j) in scenario s.

The bipartite implication relation can be expressed as *if a train arrives at a station at time $d'' + r''$ and it is planned to depart at time d with running time r, we forecast that the train will depart at time d' with running time r'*. For each leg (i,j) and each scenario s, there are three binary vectors $x_{ij} \in \{0,1\}^{|C_{ij}|}$, $y_{ij-1} \in \{0,1\}^{|C_{sij-1}|}$, and $y_{ij} \in \{0,1\}^{|C_{sij}|}$ with multiple-choice constraints for which a relation matrix can be set up. For one leg (i,j) and one scenario s, the relation matrix M^{sij} is similar to Figure 7.

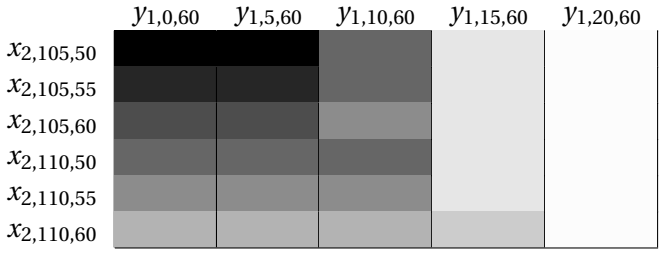


Figure 7: Example relation matrix structure for one leg and one scenario. Equal indices marked by the same gray tone.

The observable L-shaped structure for equal indices holds for every instance. For each index l in M^{sij} there are at most two blocks in M^{sij} which contain l and these blocks contain all l in M^{sij}. This property makes the following preprocessing step feasible.

Preprocessing In the preprocessing for the scenario instances, we remove all McCormick constraints from $S(M^{sij})$ and replace them by at most two 1-block inequalities. These blocks can be constructed such that they contain exactly one index and the union of these blocks form M^{sij}. In this way, we can both reduce the number of constraints in $S(M^{sij})$ and strengthen the formulation.

Instances The computations were performed on 60 instances of timetables grouped into 6 instance configurations. All instances are generated from real-world data provided by our partners at VAG, the operator of public transport in the city of Nuremberg, Germany. The names of the instance configurations follow the scheme $dy|dt|ss|nt|sn$, where $dy \in \{s, w\}$ indicates whether the timetable is for Sundays (s) or weekdays (w). For each leg in the timetable model, dt is the maximum time by which the departure time can be delayed or advanced, ss is the step size in the resulting time interval, and nt is the number of possible running times. The number of included scenarios is given by sn. The scenarios are generated from real-world tests conducted at VAG in Nuremberg, where a scenario represents the deviation of observed departure and running times from the planned ones on a single test day. We tested on 5 days (two Sundays and three weekdays), which limits the number of available scenarios. For each instance configuration, we test 10 different time horizons throughout the day, with each time horizon spanning 30 minutes. To obtain instances that can be solved to optimality within the given time limit of 10 hours by at least one of the considered methods, we optimize over one subnetwork of the train system containing 13 line segments.

Config	Con ORI	Con PRE	Sep SEP	Sep PRE+SEP
s|10|5|2|2	19450	3124	671	319
s|12|3|1|2	27397	3568	821	365
s|12|6|4|2	80392	6347	726	339
s|15|3|3|2	170094	8321	1409	814
s|18|2|1|2	108811	6929	4039	2129
w|10|5|1|3	37217	8637	8344	4769

Table 8: FRSP test: Number of constraints which are added before and after the solution process starts and percentage of separated cuts which are facets of $BIP(M)$.

Config	ORI	PRE	PRE+SEP	Cuts=0	SEP	x Factor
s|10|5|2|2	447.4	24.1	24.6	15.4	60.5	29.0
s|12|3|1|2	2272.2	137.7	47.3	37.8	131.9	60.1
s|12|6|4|2	16722.9	465.7	112.5	142.5	180.6	148.7
s|15|3|3|2	33910.7	5176.3	437.9	604.3	936.0	77.4
s|18|2|1|2	25010.6	2453.8	485.9	593.8	1278.1	51.5
w|10|5|1|3	17756.2	1337.9	740.0	1004.8	2314.6	24.0

Table 9: FRSP test: Geometric mean runtime solving to optimality.

Computational Results For each test instance, we compare five solution configurations. *ORI* is the model without n-block inequalities. For *PRE* the preprocessing step described above is applied. Additional to the preprocessing for *PRE+SEP* the separation algorithm is performed. A variant of *PRE+SEP* where we only use separated n-block inequalities and disallow Gurobi to use other cut types is carried out in *Cuts=0*. In *SEP* n-block inequalities are only separated during the solution process but no preprocessing was performed. We separate via a Gurobi callback at each node in the branch-and-bound tree one maximally violated normalized n-block constraint for each bipartite implication substructure in the problem if the violation is greater or equal 0.1. The number of n-block inequalities added to the model is presented in Table 8. It presents the mean values for each instance configuration of constraint counts. *Con ORI* and *Con PRE* denote the number of constraints in the model after Gurobi presolve without and with the inclusion of preprocessed cuts, respectively. *Sep SEP* and *Sep PRE+SEP* represent the counts of constraints added as user cuts during the solution process without and with preprocessing.

To evaluate the impact of the n-block inequalities discovered in this paper on the solution performance we compare the time the Gurobi solver takes to solve the instances to optimality and, since this may be interesting from a practical point of view, to a MIP optimality gap of 1%. This was enough time to solve each instance to optimality in at least one solution configuration. Tables 9 and 10 show for each instance configuration and each solution configuration the geometric mean of the runtime to optimality and to a MIP optimality gap of 1%, respectively. The column *x Factor* is the impact indicator and represents the factor by which the runtime of *ORI* could be shortened by the separated cuts. If for an instance the solver did not reach the demanded gap in under 10 hours it was counted as 10 hours. The number of instances which could be solved to optimality is presented in Table 11 for each instance configuration and each solution configuration.

Results Analysis The special structure in the relation matrices seems to be very suitable for the application of n-block inequalities. Preprocessing 1-block inequalities reduced the number of constraints after Gurobi presolve by more than 75%, for 15|3|3|2 by 95% on average. Although the constraint matrix in this new formulation is more densely filled, it results in much shorter runtimes of *PRE* compared to *ORI*. All of the constraints separated as user cuts in a Gurobi callback were 1-block inequalities. This is

Config	ORI	PRE	PRE+SEP	Cuts=0	SEP	x Factor
s|10|5|2|2	85.0	4.3	6.5	5.9	34.0	19.9
s|12|3|1|2	171.0	10.7	12.0	7.6	69.0	22.6
s|12|6|4|2	396.2	81.7	55.8	72.6	116.6	7.1
s|15|3|3|2	3451.8	267.6	187.9	143.6	554.1	24.0
s|18|2|1|2	3545.0	201.7	106.2	136.1	371.6	33.4
w|10|5|1|3	547.4	91.1	53.1	103.2	189.7	10.3

Table 10: FRSP test: Geometric mean runtime solving to a MIP optimality gap of 1%.

Config	ORI	PRE	PRE+SEP	Cuts=0	SEP
s|10|5|2|2	10/10	10/10	10/10	10/10	10/10
s|12|3|1|2	10/10	10/10	10/10	10/10	10/10
s|12|6|4|2	8/10	10/10	10/10	10/10	10/10
s|15|3|3|2	1/10	10/10	10/10	10/10	10/10
s|18|2|1|2	2/10	9/10	10/10	10/10	9/10
w|10|5|1|3	5/10	10/10	10/10	10/10	10/10

Table 11: FRSP test: Number of instances which were solved to optimality in under 10 hours.

due to the special block structure in the relation matrix. However, the conclusion that this observation suggests, namely that $BIP(M)$ for a relation matrix M with the above mentioned L-shaped structure can only be described by bounds and 1-block inequalities, is wrong, as the following counterexample shows.

Example 3. Consider the relation matrix

$$M = \begin{pmatrix} 1 & 2 & 3 & 5 \\ 2 & 2 & 4 & 5 \\ 1 & 3 & 3 & 5 \\ 4 & 5 & 4 & 5 \end{pmatrix}.$$

The 2-block constraint

$$2x_1 + x_2 + 2x_3 + x_4 + 2y_1 + y_2 + 2y_3 \leq 2z_1 + z_2 + 2z_3 + z_4 + 2$$

is a facet inducing inequality for $BIP(M)$.

Yet, the separated 1-block inequalities seem to be very effective in closing the dual bound. Due to the quickness of the separation LP, frequently calling the separation routine does not have a negative effect on the runtime. Comparing *ORI* and *SEP* we observe a constant improvement across all instance configurations by this separation. Combining the preprocessing and the separation routine we observe a significant impact of n-block inequalities to the solution of the scenario timetable models both to optimality and to a MIP gap of 1%. *PRE+SEP* in contrast to *ORI* was able to solve all tested instances to optimality. Particularly impressive is the difference in the number of solved instances in the configuration 15|3|3|2. While the model without n-block inequalities could not be solved to optimality after 10 hours in 90% of the instances, the geometric mean runtime of *PRE+SEP* was 437.9 seconds. In a little less than 1 hour, Gurobi was able to reduce the MIP gap to 1% but was not able to close the dual bound further in the next 9 hours. Here the separation of 1-block inequalities turned out to be crucial. Setting the Gurobi parameter *Cuts* to 0 and with that disallowing any other cut class than n-block inequalities to be separated did improve the runtime to optimality in 2 of the 6 test configurations. The runtime to a MIP gap of 1% was improved in half of the instance configurations. Overall these classical cut classes like MIR, RLT or BQP cuts did not have a major impact on the solution performance when n-block inequalities were added.

Evaluation on Realistic Problem Sizes To demonstrate the practical applicability of our approach, we evaluate its performance on large-scale instances that reflect realistic problem sizes. The previous experiments were conducted on a subnetwork of 13 line segments to ensure tractability. In this section, we test instances on the full network of the two autonomously driving lines in Nuremberg, which consists of 46 line segments. Additionally, we evaluate scalability with respect to the number of scenarios by increasing from 2 to 10 scenarios for Sunday timetables (the additional 8 scenarios are randomly generated). For these large-scale tests, we compare *ORI* and *PRE+SEP*, the best-performing method from the previous experiments. We again report results over 10 different time horizons of 30 minutes each. Table 12 presents the geometric mean runtime (instances that reached the time limit of 10 hours are set to 36000), the mean optimality gap after 10 hours, and the number of instances solved to optimality.

Config	Method	Solved	Runtime (s)	Gap (%)
10 scenarios	ORI	4/10	29321.2	
s|10|5|2|10	PRE+SEP	10/10	603.9	
Full network	ORI	0/10		3.53
s|10|5|2|2	PRE+SEP	1/10		0.54

Table 12: Large-scale FRSP test: Comparison of *ORI* and *PRE+SEP* on full network and increased scenario counts.

The results demonstrate that n-block inequalities are essential for solving realistic problem sizes. For 10 scenarios, *PRE+SEP* solves all instances to optimality, whereas *ORI* solves only 40% within the time limit, with a geometric mean runtime nearly 50 times slower. On the full network, both methods struggle, but *PRE+SEP* achieves a substantially better mean optimality gap of 0.54% compared to 3.53% for *ORI*. With *PRE+SEP*, all 10 instances reached an optimality gap of 1% in under 10 hours, while none did with *ORI*. These results confirm that the developed preprocessing and separation routines enable the solution of operationally relevant instances that would otherwise be intractable.

6.3 Application to the Quadratic Assignment Problem

Our final study investigates the broader applicability of n-block inequalities to problems where the chaining of BIP structures contains cycles. While we can no longer guarantee a complete convex hull description in this case, our cutting planes remain valid and can still be highly effective. The QAP provides an excellent benchmark to evaluate their practical strength in such general settings.

Koopmans and Beckmann presented a quadratic integer formulation for the quadratic assignment problem in [26]. In their application case, they aim to optimize the allocation of a set of m plants to m specific locations, modeled by binary variables $x \in \{0,1\}^{m\times m}$. The objective is to minimize the total cost, which combines distance-based costs, flow-based costs, and placement costs. Mathematically, it involves three input matrices representing commodity flows between facilities ($F \in \mathbb{R}_+^{m\times m}$), distances between locations ($D \in \mathbb{R}_+^{m\times m}$), and placement costs ($B \in \mathbb{R}_+^{m\times m}$). The quadratic integer model becomes

$$\begin{aligned}
\min \quad & \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m}\sum_{l=1}^{m} f_{ij}x_{ik}d_{kl}x_{jl} + \sum_{i,j=1}^{m} b_{ij}x_{ij} && \text{(QAP)}\\
\text{s.t.} \quad & \sum_{i=1}^{m} x_{ij} = 1, \quad \forall j \in [m] \\
& \sum_{j=1}^{m} x_{ij} = 1, \quad \forall i \in [m] \\
& x_{ij} \in \{0,1\}, \quad \forall ij \in [m]^2.
\end{aligned}$$

We can reformulate (QAP) into a BIP based model as follows. Define

$$X^i := \{i_1 i_2 \in [m]^2 \mid i_1 = i\} \text{ for all } i \in [m] \text{ and } Y^j := \{j_1 j_2 \in [m]^2 \mid j_2 = j\} \text{ for all } j \in [m].$$

We can group pairs of elements $ii_2 \in X^i$ and $j_1 j \in Y^j$ with identical costs $f_{ii_2} d_{j_1 j}$ together and introduce a variable z_l^{ij} for each cost group $l \in Z^{ij}$ with corresponding costs $\tilde{c}_l^{ij}$. For each $ij \in [m]^2$, we define a function $f^{ij} : [m]^2 \to Z^{ij}$ which maps $i_2 j_1$ to the cost group of $ii_2 j_1 j$ for each pair of elements $ii_2 \in X^i$ and $j_1 j \in Y^j$. This yields an equivalent formulation of (QAP):

$$\begin{aligned}
\min \quad & \sum_{i\in[m]} \sum_{j\in[m]} \sum_{l\in Z^{ij}} \tilde{c}_l^{ij} z_l^{ij} + \sum_{i,j=1}^{m} b_{ij} x_{ij} && \text{(ITQAP)}\\
\text{s.t.} \quad & \sum_{ii_2 \in X^i} x_{ii_2} = 1, \quad \forall i \in [m] \\
& \sum_{j_1 j \in Y^j} x_{j_1 j} = 1, \quad \forall j \in [m] \\
& \sum_{l\in Z^{ij}} z_l^{ij} = 1, \quad \forall (i,j) \in [m]^2 \\
& x_{ii_2} x_{j_1 j} \le z^{ij}_{f^{ij}(i_2 j_1)}, \quad \forall (i, i_2, j_1, j) \in [m]^4 \\
& x_{ij} \in \{0,1\}, \quad \forall (i,j) \in [m]^2.
\end{aligned}$$

Here, we can directly observe an bipartite implication instance with relation matrix M^{ij}, where $M^{ij}_{i_2 j_1} := f^{ij}(i_2, j_1)$ for $i_2 \in [m]$ and $j_1 \in [m]$ as a substructure of (QAP) for each $ij \in [m]^2$. The chaining of these instances differs from the one observed in Section 5.

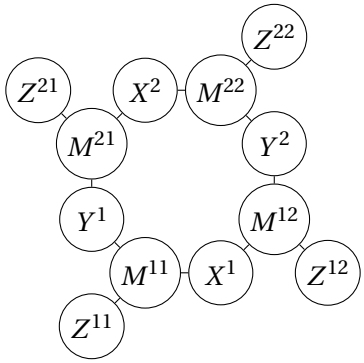


Figure 8: Dependency graph for the chaining of bipartite implication instances of Model (QAP) with $n = 2$.

As Figure 8 shows, the dependency graph of the CPMC extension for (QAP) is not a forest, therefore n-block inequalities for the substructured BIPs are not necessarily sufficient to define the convex hull of the feasible points of Model (QAP). Still, these n-block inequalities are valid and lead to significant improvements for the solution process of (ITQAP), as the subsequent computational study demonstrates.

Note, that there have been many strong improvements to the linearization of the Koopmans and Beckmann formulation, in which the BIP is not directly applicable [46]. When combined with problem-specific solving strategies, these improvements render state-of-the-art QAP solvers far superior to methods based on the linearization of the Koopmans and Beckmann formulation, even when enhanced by the separation of n-block inequalities. Nevertheless, comparing the performance of this formulation with and without the separation routine provides a valuable way to highlight the practical effectiveness of the cutting planes, particularly in scenarios where multiple bipartite implication substructures exist that do not form a simple chained tree structure.

Config	$\gamma/(\alpha\cdot\beta)$	Sep Provided	Sep Used	ORIGINAL	BIP
chr18b	0.0357	1841	1333	5.1%	**124.1**
nug16b	0.0606	557850	55105	61.2%	**5.1%**
nug16a	0.0693	405074	66239	81.2%	**8.9%**
nug15	0.0712	239121	38906	52.1%	**22155.7**
nug14	0.0781	222419	30857	64.2%	**15185.7**
scr20	0.0826	86759	29343	29.9%	**8.0%**
had20	0.0867	66820	28312	96.0%	**14.3%**
chr18a	0.0959	16026	11053	27207.8	**1417.5**
had18	0.0973	221810	25076	94.6%	**11.0%**
nug12	0.0979	13508	6231	16053.8	**439.3**
had16	0.1127	190317	22271	87.7%	**6.8%**
scr15	0.1134	5741	3042	6359.4	**762.6**
chr15a	0.1141	5717	5	802.5	**236.3**
chr15b	0.1141	4602	590	304.4	**192.0**
chr15c	0.1141	5777	574	**122.2**	179.5
scr12	0.1375	2692	1426	207.1	**83.5**
had14	0.1396	60990	11162	67.9%	**10607.3**
had12	0.1525	65653	7391	41.3%	**2649.1**
lipa20b	0.2080	86306	11493	94.0%	**2.6%**
tai15b	0.2893	112754	12128	0.6%	**29469.7**
tai12b	0.3888	59506	6771	**4167.9**	7806.2
tai10b	0.4047	4273	3330	**67.0**	182.9
tai10a	0.6558	9824	2830	**1803.9**	2385.6
tai12a	0.6934	158225	3529	**17.5%**	22.2%
rou20	0.7010	12756	12204	**94.4%**	100.0%
tai15a	0.7133	27661	6482	**76.2%**	99.9%
rou15	0.7472	31620	8878	**71.1%**	88.4%
rou12	0.7645	7772	3517	**22.9%**	39.6%

Table 13: QAP study results: z-ratio $\gamma/(\alpha\cdot\beta)$, number of separated cuts (*Sep Provided*), number of cuts used by Gurobi (*Sep Used*), runtime/MIP optimality gap after 10 hours without separation (*ORIGINAL*) and with separation (*BIP*).

Instances We analyze 28 instances from the well established QAPLIB [6]. Note that the number in the name of the instances equals the parameter m in Model (ITQAP).

Computational Results We solve each instance both with and without the use of the separation algorithm for normalized n-block inequalities described in Section 4.2. At each node in the branch-and-bound tree, we collect the maximally violated cut in each BIP subproblem. All cuts with violations greater than or equal to 0.01 and at least 10% of the maximum observed violation at the node are then passed to Gurobi as UserCuts. Gurobi then decides, whether to add the cut to the model. We omitted a comprehensive analysis of the relation matrix which could be used to add instance-adapted constraints to the model in preprocessing in order to show the performance of the separated cuts on general QAP instances. The precomputing of cuts using the clique technique described in Section 4.3 was also not carried out, because the z-ratio, i.e., the ratio of the number of z-indices (γ) to the matrix size ($\alpha\cdot\beta$) was too big, as we can see in Column $\gamma/(\alpha\cdot\beta)$ of Table 13.

All instances were solved with a time limit of 10 hours. Column *Sep Provided* of Table 13 shows the number of separated n-block inequalities for each instance. *Sep Used* displays the number of cuts which were added to the model by Gurobi. The runtime to optimality or the relative MIP optimality gap

in case of the time limit being exceeded for the model with (*BIP*) and without (*ORIGINAL*) separated n-block inequalities are also displayed in Table 13. The shorter runtime or smaller optimality gap are marked in bold.

Results Analysis We sorted Table 13 by the z-ratio to illustrate the strong correlation of this parameter with the positive impact of the separated n-block inequalities. For low values of $\gamma/(\alpha \cdot \beta)$, the separation of n-block inequalities yields a significant improvement in reducing the relative MIP optimality gap and shortens the runtime drastically. The fast runtime of the separation LP enables to add a large number of inequalities which help the solver to cut off non-integer solutions. However, higher z-ratios worsen the performance of the separated cuts. They can even lead to higher MIP optimality gaps. The numerical results in Table 13 indicate a positive effect of n-block inequalities for z-ratios up to about 0.3. The results are consistent with the observations in Section 6.1. Low z-ratios lead to potentially larger 1-blocks when fixing the z-indices in the block. These 1-blocks can be utilized to form tight n-block inequalities. In contrast to the FRSP case study, in the QAP study not only 1-block inequalities were separated, but also blocks for higher values of $n \leq 80$, even though 1-block inequalities make up the largest proportion at around 90%.

6.4 Summary and Practical Recommendations

Our computational experiments validate the theoretical framework and demonstrate that the separation of n-block inequalities is computationally efficient, scaling well with instance size. For tree-like chained bipartite implication substructures, as demonstrated in the fixed recourse stochastic programming application, our cutting-plane approach establishes a new state-of-the-art with significant improvements in solution times. Moreover, the method remains effective even for general chaining structures, as evidenced by the improvements on quadratic assignment problem instances, indicating broader applicability.

Based on our numerical results, we offer practical guidance for practitioners. The ratio of z-variables to the matrix size serves as a key indicator, with problems where this ratio is small showing the most significant benefits. An important finding is that one-block inequalities constitute the vast majority of effective cuts across all test cases. For most practical applications, restricting separation to one-block inequalities, particularly those involving few z-variables, provides substantial improvements with minimal computational overhead. When the relative number of z-indices is low, preprocessing with the clique-based algorithm from Section 4.3 can dramatically reduce the number of constraints. We recommend implementing separation as a callback within the branch-and-cut framework. In summary, n-block inequalities provide a powerful tool for optimization problems with bipartite implication structures, with one-block inequalities alone offering an accessible and effective approach for a wide range of applications.

7 Further Applications

The computational study showed that separating n-block inequalities can substantially improve the runtime of problems with bipartite implication substructure. Therefore, we want to explore further applications with bipartite implication substructure that are promising to analyze in future studies.

7.1 Stepwise Interpolation of Non-Linear Functions

While stepwise interpolations provide a natural way to handle non-linear equations in engineering applications, the resulting binary formulations are rarely used in practice due to the computational burden of introducing many binary variables. However, our computational results suggest that the strengthened formulations obtained through BIP theory could make binary modeling approaches competitive again. The significant improvements in LP relaxation quality and solution times we

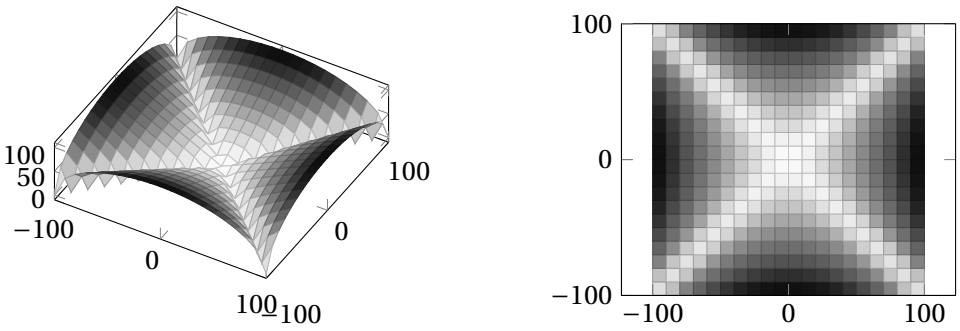


Figure 9: Piecewise constant approximation of the function $p_E^2 - p_A^2 = |q|q$

observed might outweigh the increased model size. In many engineering applications, non-linear equations appear in the constraints. One possibility to deal with the complexity of these non-linearities is to approximate them by staircase functions. Let's consider a general nonlinear equation:

$$f(x_1, x_2, ..., x_n) = 0 \tag{25}$$

where f is any nonlinear function with n variables. A classic example from gas network optimization is the Weymouth equation [24]

$$p_E^2 - p_A^2 = |q|q, \tag{26}$$

which describes the pressure loss within a pipeline, i.e., the relationship between the pressure at the inlet p_E and outlet p_A of a pipe section and the transport volume q. This equation can be approximated by discretizing both the pressure domain (p_E and p_A) and the flow domain (q) into intervals, as illustrated in Figure 9. The selection of intervals for the inlet and outlet pressures then implies a specific interval for the flow value according to the physical equation.

For equations involving exactly three variables, like this gas network example, we can directly apply the BIP theory developed in this paper. After discretizing each variable's domain into intervals, two variables can be considered as the implying variables and one as the implied variable. The relation matrix M contains the mapping between interval combinations of the two implying variables to the intervals of the implied variable based on the physical equation. For equations involving more than three variables or multiple nonlinear terms, we need to extend the bipartite implication concept to multiple implying sets.

7.2 Piecewise Linear Relaxations for MINLP

Another very promising candidate for applying BIP theory are piecewise linear relaxations for mixed-integer nonlinear programming (MINLP). The multiple-choice method [1] for piecewise linear approximations divides the domain of each variable into intervals and introduces binary variables to indicate which interval is active. For a function $f(x)$, let $\{x^0, \ldots, x^n\}$ be the breakpoints of the intervals. The continuous variable x is then expressed as

$$\begin{aligned} x &= \sum_{i=1}^{n} \lambda_i, \\ \sum_{i=1}^{n} y_i &= 1, \\ y_i x^{i-1} \le \lambda_i &\le y_i x^i \quad \forall i \in [n], \\ y_i &\in \{0,1\} \quad \forall i \in [n], \end{aligned}$$

where y_i are the binary interval indicator variables and λ_i is the residual in interval i for $i \in [n]$. When applying this method to relations between 3 continuous variables in the model, a bipartite implication

structure naturally emerges, though with a slight modification from our basic theory. These relations do not need to be explicitly expressed as equations in the original model. When using expression trees to reformulate the problem such that it contains only one-dimensional non-linear functions, relations between both original variables and artificially created variables emerge naturally. Consider the constraint $\sin(xy) = 0$: we introduce a new variable z and add the constraints $z = xy$, $\sin(z) = 0$. After discretization using the multiple-choice method, each variable's domain is partitioned, and we introduce binary variables for the intervals of x, y, and z.

The key difference from our basic BIP theory is that selecting specific intervals i and j for x and y does not necessarily determine a unique interval l for z. Instead, it restricts z to a subset of possible intervals, depending on the actual values chosen within the intervals of x and y. This means the relation matrix M would need to map each pair (i, j) to a set of possible l values rather than a single value.

To properly apply our BIP theory in this context, we would need to extend it to allow for set-valued entries in the relation matrix M, where M_{ij} represents the set of possible intervals for z when x is in interval i and y is in interval j. This would require adapting the theory of n-block inequalities to handle these set-valued relationships.

Over the past two decades, piecewise linear approximations have been extensively studied in the literature, establishing them as a viable alternative to spatial branching methods for solving MINLPs, see [14, 22]. The multiple-choice method described here is one of several modeling techniques, but it is rarely used in practice due to its weak LP relaxation bounds. The more commonly applied incremental method [31], which is also directly compatible with our BIP framework (as discussed in Appendix A), often yields tighter relaxations. While a systematic exploration of these methods in the context of BIP is beyond the scope of this paper, our preliminary results suggest that integrating BIP-based cuts into piecewise linear relaxation techniques has the potential to significantly improve computational performance, particularly in cases where the resulting BIP substructures exhibit favorable properties (e.g., low z-ratios and tree-like dependencies).

7.3 Scenario Expansion for Bayesian Networks

Bayesian networks provide a powerful framework for modeling probabilistic dependencies between variables. In optimization contexts, some nodes in these networks may represent decision variables rather than random variables. Consider a discrete Bayesian network where one of the parent nodes represents our decision variable (with states that are scenario-independent), while the other parent and the child node represent uncertain outcomes.

An illustrative example appears in healthcare resource planning: A hospital needs to decide on staffing levels (low/medium/high). This is our decision variable represented by the first parent node, whose state choice remains constant across all scenarios. The second parent node might represent patient arrival rates (low/medium/high), and the child node represents quality of care (sufficient/insufficient).

When generating scenarios, we sample from the conditional probability tables of the Bayesian network. For each scenario s, this sampling process determines exactly which state of the child node will occur for each combination of parent node states - there is no probability involved anymore within a single scenario. This deterministic relationship for each scenario s is captured by the relation matrix M^s. In other words, sampling converts the probabilistic Bayesian network into a set of deterministic bipartite implication relationships, one for each scenario.

For a basic two-parent network structure (which can represent any tree network through divorcing as described in [34]), let:

- $\mathscr{X} = \{1, \ldots, n_x\}$ be the states of the decision parent node (scenario-independent)
- $\mathscr{Y} = \{1, \ldots, n_y\}$ be the states of the uncertain parent node
- $\mathscr{Z} = \{1, \ldots, n_z\}$ be the states of the child node

- $s \in \mathscr{S}$ be the set of scenarios

We introduce binary variables:

$$x_i = \begin{cases} 1 & \text{if decision parent is in state } i \\ 0 & \text{otherwise} \end{cases}$$

$$y_{js} = \begin{cases} 1 & \text{if uncertain parent is in state } j \text{ in scenario } s \\ 0 & \text{otherwise} \end{cases}$$

$$z_{ls} = \begin{cases} 1 & \text{if child is in state } l \text{ in scenario } s \\ 0 & \text{otherwise} \end{cases}$$

Each scenario s generates its own relation matrix M^s, where M^s_{ij} indicates which state in the child node becomes active when the decision parent is in state i and the uncertain parent is in state j. The sampling process thus does not directly determine the states of the nodes, but rather shapes how the states of the parent nodes relate to the states of the child node in each scenario. This creates a natural bipartite implication structure per scenario: if we choose a certain staffing level (scenario-independent) and a certain arrival rate occurs in scenario s, then this implies a specific quality of care level according to the relation matrix M^s for that scenario.

This concept allows us to apply the BIP theory to a wide range of practical problems involving decision-making under uncertainty, where the relationships between decisions and outcomes may vary across scenarios.

8 Conclusion

In this article, we introduced the bipartite implication polytope (BIP), a special case of the bipartite quadric polytope that models conditional relations across three sets of binary variables, where selections within two implying sets imply a choice in a corresponding implied set. We provided the complete description of the polytope using solely newly defined and characterized n-block inequalities and bound constraints. Additionally, we showed how to separate these n-block inequalities in polynomial time and presented a routine to efficiently precompute tight 1-block inequalities if the structure of the relation matrix is known.

Our computational study demonstrated the practical value of the BIP theory through two distinct applications: First, in fixed recourse stochastic programming, specifically for railway timetabling. Here, the present tree-like chaining of the substructured BIPs caused significant runtime improvements in a real-world application. Second, through the quadratic assignment problem, where - while not competing with specialized QAP solvers - we demonstrated how our separated cuts can improve the solution process for problems with arbitrary chained bipartite implication substructures. This twofold validation underscores the broad applicability of our approach. Beyond these applications, we explored several other promising fields including stepwise interpolation of non-linear functions, piecewise linear relaxations for MINLP, and scenario expansion for Bayesian networks.

Overall, this work provides a deeper insight into the structure of binary quadratic problems with multiple-choice constraints and a new approach to efficient optimization over the BIP. However, there is still significant potential for further research. On the theoretical side, two main extensions of the BIP theory appear particularly promising: First, allowing multiple implying sets instead of just two, and second, extending the theory to handle set-valued entries in the relation matrix where one combination of implying variables implies a set of possible implied variables. These extensions would be particularly valuable for applications in stepwise interpolation and piecewise linear relaxations of nonlinear functions.

Additionally, the chaining of relation matrices that was present in the stochastic railway timetabling model can be extended to other tree-like structures. With regard to possible applications, we see a wide

range even beyond the areas addressed in this paper. The approach presented here can therefore serve as a foundation for developing more general theoretical frameworks to handle complex relationships in discrete optimization problems.

Acknowledgements

We wish to acknowledge the significant contribution of Jan Krause, who provided invaluable assistance with proofreading. His attention to detail has greatly enhanced the quality of our work.

Funding: This work was part of the Project EKSSE, funded by the Bundesministerium für Bildung und Forschung (BMBF).

A Bipartite Implications in the Incremental Method

The incremental method for piecewise linear approximations, introduced in [31], models the domain of a variable by sequentially activating segments between breakpoints. For a function $f(x)$ with breakpoints $\{x^0, \dots, x^n\}$, the method introduces continuous variables $\delta_i \in [0,1]$ for $i \in [n]$ and binary variables $z_i \in \{0,1\}$ for $i \in [n-1]$. The variable x is expressed as an incremental combination of the segments:

$$x = x^0 + \sum_{i=1}^{n} \delta_i \left(x^i - x^{i-1} \right),$$

subject to constraints that enforce the sequential activation of segments:

$$\begin{aligned} \delta_1 &\leq 1, \\ \delta_{i+1} \leq z_i \leq \delta_i &\quad \forall i \in [n-1], \\ \delta_n &\geq 0, \\ z_i \in \{0,1\} &\quad \forall i \in [n-1]. \end{aligned}$$

Here, δ_i represents the fraction of the segment between x^i and x^{i+1} that is utilized. The binary variable z_i ensures that segment $i+1$ can only be active if segment i is fully traversed ($\delta_i = 1$). This formulation guarantees that at most one δ_i is fractional, maintaining a single active segment in the approximation.

To utilize the developed BIP theory, we need sets of binary variables with multiple-choice constraints. This can be achieved by mapping the z-variables to a set of binaries $y_i \in \{0,1\}$ for $i \in [n-1]$, setting

$$\begin{aligned} y_1 &:= 1 - z_1, \\ y_i &:= z_{i-1} - z_i \quad \forall i \in \{2, ..., n-1\}, \\ y_n &:= z_{n-1}. \end{aligned}$$

As in the multiple-choice method these variables indicate which interval between two breakpoints is active. Hence, a multiple choice constraint on this set of binaries is implied. Again, if relations between three of the continuous variables exist, a bipartite implication structure emerges.

The presented mapping from activation indicator variables z to segment indicator variables y is bijective, therefore after separating n-block inequalities on the y-variables, we can re-substitute the z-variables in the obtained cut, which is then feasible for the original problem.

References

[1] Anantharam Balakrishnan and Stephen C. Graves. A composite algorithm for a concave-cost network flow problem. *Networks*, 19:175–202, 1989. URL `https://api.semanticscholar.org/CorpusID:8480043`.

[2] Francisco Barahona, Michael Jünger, and Gerhard Reinelt. Experiments in quadratic 0–1 programming. *Mathematical Programming*, 44(1):127–137, 1989.

[3] Endre Boros and Peter L Hammer. Cut-polytopes, boolean quadric polytopes and nonnegative quadratic pseudo-boolean functions. *Mathematics of Operations Research*, 18(1):245–253, 1993.

[4] Endre Boros and Peter L Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1-3):155–225, 2002.

[5] Rainer E. Burkard, Eranda Çela, Panos M. Pardalos, and Leonidas S. Pitsoulis. The quadratic assignment problem. *Handbook of Combinatorial Optimization*, pages 1713–1809, 1998. doi: 10.1007/978-1-4613-0303-9_27. URL https://link.springer.com/chapter/10.1007/978-1-4613-0303-9_27.

[6] RE Burkard, E Çela, SE Karisch, F Rendl, M Anjos, and P Hahn. QAPLIB - a quadratic assignment problem library - problem instances and solutions. https://doi.org/10.7488/ds/3428, 2022. [dataset].

[7] Andreas Bärmann, Patrick Gemander, and Maximilian Merkert. The clique problem with multiple-choice constraints under a cycle-free dependency graph. *Discrete Applied Mathematics*, 284, 2020.

[8] Andreas Bärmann, Alexander Martin, and Oskar Schneider. The bipartite boolean quadric polytope with multiple-choice constraints. *arXiv: Optimization and Control*, 2020.

[9] Andreas Bärmann, Patrick Gemander, and Alexander Martin. A stochastic optimization approach to energy-efficient underground timetabling under uncertain dwell and running times. *Optimization Online*, 2022.

[10] Alberto Ceselli, Roberto Cordone, Yari Melzani, and Giovanni Righini. Optimization algorithms for the max edge weighted clique problem with multiple choice constraints. In *Proceedings of the 9th Cologne-Twente Workshop on Graphs and Combinatorial Optimization*, pages 37–41, 2010.

[11] Wen-Chieh Chang, Sudheer Vakati, Roland Krause, and Oliver Eulenstein. Exploring biological interaction networks with tailored weighted quasi-bicliques. In *BMC Bioinformatics*, volume 13, pages 1–9. Springer, 2012.

[12] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4(4):305–337, 1973. ISSN 0012-365X. doi: https://doi.org/10.1016/0012-365X(73)90167-2. URL https://www.sciencedirect.com/science/article/pii/0012365X73901672.

[13] Vašek Chvátal. On certain polytopes associated with graphs. *Journal of Combinatorial Theory*, 18:138–154, 1975.

[14] Keely L. Croxton, Bernard Gendron, and Thomas L. Magnanti. A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems. *Management Science*, 49(9):1268–1273, 2003. doi: 10.1287/mnsc.49.9.1268.16570.

[15] Abraham Duarte, Manuel Laguna, Rafael Martí, and Jesús Sánchez-Oro. Optimization procedures for the bipartite unconstrained 0-1 quadratic programming problem. *Computers & Operations Research*, 51:123–129, 2014.

[16] Richard J Forrester and Noah Hunt-Isaak. Computational comparison of exact solution methods for 0-1 quadratic programs: Recommendations for practitioners. *Journal of Applied Mathematics*, 2020:1–21, 2020.

[17] A. M. Geoffrion and G. W. Graves. Scheduling parallel production lines with changeover costs: Practical application of a quadratic assignment/lp approach. *https://doi.org/10.1287/opre.24.4.595*, 24:595–610, 8 1976. ISSN 0030364X. doi: 10.1287/OPRE.24.4.595. URL https://pubsonline.informs.org/doi/abs/10.1287/opre.24.4.595.

[18] Fred Glover. Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22(4):455–460, 1975.

[19] Fred Glover and Eugene Woolsey. Converting the 0-1 polynomial programming problem to a 0-1 linear program. *Operations Research*, 22(1):180–182, 1974.

[20] Fred Glover, Tao Ye, Abraham P Punnen, and Gary Kochenberger. Integrating tabu search and vlsn search to develop enhanced algorithms: A case study using bipartite boolean quadratic programs. *European Journal of Operational Research*, 241(3):697–707, 2015.

[21] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023. URL `https://www.gurobi.com`.

[22] Joey Huchette and Juan Pablo Vielma. Nonconvex piecewise linear functions: Advanced formulations and simple modeling tools. *Operations Research*, 72(5):1835–1856, 2022. doi: https://doi.org/10.1287/opre.2019.1973.

[23] Daniel Karapetyan, Abraham P Punnen, and Andrew J Parkes. Markov chain methods for the bipartite boolean quadratic programming problem. *European Journal of Operational Research*, 260(2):494–506, 2017.

[24] Thorsten Koch, Benjamin Hiller, Marc E. Pfetsch, and Lars Schewe. *Evaluating Gas Network Capacities*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2015. doi: 10.1137/1.9781611973693. URL `https://epubs.siam.org/doi/abs/10.1137/1.9781611973693`.

[25] Gary Kochenberger, Jin-Kao Hao, Fred Glover, Mark Lewis, Zhipeng Lü, Haibo Wang, and Yang Wang. The unconstrained binary quadratic programming problem: a survey. *Journal of Combinatorial Optimization*, 28:58–81, 2014.

[26] Tjalling C. Koopmans and Martin Beckmann. Assignment problems and the location of economic activities. *Econometrica*, 25(1):53–76, 1957. ISSN 00129682, 14680262. URL `http://www.jstor.org/stable/1907742`.

[27] Adam N Letchford and Michael M Sørensen. A new separation algorithm for the boolean quadric and cut polytopes. *Discrete Optimization*, 14:61–71, 2014.

[28] Leo Liberti. Compact linearization for binary quadratic problems. *4OR*, 5(3):231–245, 2007.

[29] Ricardo M Lima and Ignacio E Grossmann. On the solution of nonconvex cardinality boolean quadratic programming problems: a computational study. *Computational Optimization and Applications*, 66:1–37, 2017.

[30] Guimei Liu, Kelvin Sim, and Jinyan Li. Efficient mining of large maximal bicliques. In *Data Warehousing and Knowledge Discovery: 8th International Conference, DaWaK 2006, Krakow, Poland, September 4-8, 2006. Proceedings 8*, pages 437–448. Springer, 2006.

[31] Harry M. Markowitz and Alan S. Manne. On the solution of discrete programming problems. *Econometrica*, 25(1):84–110, 1957. ISSN 00129682, 14680262. URL `http://www.jstor.org/stable/1907744`.

[32] Andrew Mason and Mikael Rönnqvist. Solution methods for the balancing of jet turbines. *Computers & Operations Research*, 24(2):153–167, 1997. ISSN 0305-0548. doi: https://doi.org/10.1016/S0305-0548(96)00047-0. URL `https://www.sciencedirect.com/science/article/pii/S0305054896000470`.

[33] George L. Nemhauser and Laurence A. Wolsey. A recursive procedure to generate all cuts for 0–1 mixed integer programs. *Mathematical Programming*, 46:379–390, 1990. URL `https://api.semanticscholar.org/CorpusID:206799928`.

[34] Kristian G. Olesen, Uffe Kjaerulff, Frank Jensen, Finn V. Jensen, Bjørn Falck, Steen Andreassen, and Stig K. Andersen. A Munin Network for the Median Nerve-a Case Study on Loops. *Applied Artificial Intelligence*, 3(2-3):385–403, January 1989. ISSN 0883-9514. doi: 10.1080/08839518908949933. URL https://doi.org/10.1080/08839518908949933. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/08839518908949933.

[35] Manfred W. Padberg. The boolean quadric polytope: Some characteristics, facets and relatives. *Mathematical Programming*, 45:139–172, 1989.

[36] Abraham P. Punnen and Yang Wang. The bipartite quadratic assignment problem and extensions. *European Journal of Operational Research*, 250(3):715–725, 2016. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2015.10.006. URL https://www.sciencedirect.com/science/article/pii/S0377221715009133.

[37] Abraham P Punnen, Piyashat Sripratak, and Daniel Karapetyan. Domination analysis of algorithms for bipartite boolean quadratic programs. In *International Symposium on Fundamentals of Computation Theory*, pages 271–282. Springer, 2013.

[38] Abraham P Punnen, Piyashat Sripratak, and Daniel Karapetyan. The bipartite unconstrained 0–1 quadratic programming problem: Polynomially solvable cases. *Discrete Applied Mathematics*, 193:1–10, 2015.

[39] Hanif D Sherali and J Cole Smith. An improved linearization strategy for zero-one quadratic programming problems. *Optimization Letters*, 1:33–47, 2007.

[40] Hanif D Sherali, Youngho Lee, and Warren P Adams. A simultaneous lifting strategy for identifying new classes of facets for the boolean quadric polytope. *Operations Research Letters*, 17(1):19–26, 1995.

[41] H.D. Sherali and W.P. Adams. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*. Nonconvex Optimization and Its Applications. Springer US, 1998. ISBN 9780792354871. URL https://books.google.fr/books?id=sPzzL4VvWqsC.

[42] Piyashat Sripratak. *The bipartite boolean quadratic programming problem*. PhD thesis, Simon Fraser University, 2014.

[43] Piyashat Sripratak, Abraham P Punnen, and Tamon Stephen. The bipartite boolean quadric polytope. *Discrete Optimization*, 44:100657, 2022.

[44] Ivar Ugi, Johannes Bauer, Josef Brandt, Josef Friedrich, Johann Gasteiger, Clemens Jochum, and Wolfgang Schubert. New applications of computers in chemistry. *Angewandte Chemie International Edition in English*, 18:111–123, 1979. ISSN 0570-0833. doi: 10.1002/ANIE.197901111. URL https://portal.fis.tum.de/en/publications/new-applications-of-computers-in-chemistry.

[45] Qinghua Wu, Yang Wang, and Fred Glover. Advanced tabu search algorithms for bipartite boolean quadratic programs guided by strategic oscillation and path relinking. *INFORMS Journal on Computing*, 32(1):74–89, 2020.

[46] Huizhen Zhang, Cesar Beltran-Royo, and Miguel Constantino. Effective formulation reductions for the quadratic assignment problem. *Computers & Operations Research*, 37(11):2007–2016, 2010. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2010.02.001. URL https://www.sciencedirect.com/science/article/pii/S0305054810000316. Metaheuristics for Logistics and Vehicle Routing.

[47] Leishan Zhou, Lu (Carol) Tong, Junhua Chen, Jinjin Tang, and Xuesong Zhou. Joint optimization of high-speed train timetables and speed profiles: A unified modeling approach using space-time-speed grid networks. *Transportation Research Part B: Methodological*, 97:157–181, 2017. ISSN 0191-2615. doi: https://doi.org/10.1016/j.trb.2017.01.002. URL https://www.sciencedirect.com/science/article/pii/S0191261516301059.

[48] Ante Ćustić, Vladyslav Sokol, Abraham P. Punnen, and Binay Bhattacharya. The bilinear assignment problem: Complexity and polynomially solvable special cases, 2016.