THE IF-THEN POLYTOPE: CONDITIONAL RELATIONS OVER MULTIPLE SETS OF BINARY VARIABLES*

ROBERT BURLACU†, PATRICK GEMANDER†, AND TOBIAS KUEN†

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 "if" sets imply a choice in a corresponding "then" set. We call this polytope the *if-then 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 if-then relations, the descriptions of the if-then 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.

Key words. 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

$$\mathrm{QP}(G) := \mathrm{conv}\{(x, z) \in \{0, 1\}^{V \cup E} \mid x_i x_j = z_{ij},\ (i, j) \in E\}$$

was introduced in [28] 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. Notably, 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. More formally, we can represent this 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

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

In practice, it is irrelevant which edge is chosen specifically; our only concern is to evaluate the cost of the edge. To facilitate this, we group edges with identical costs together, introduce a variable z_l for each group $l \in L$ and assign the corresponding costs c_l. With $f : E \to L$ as the function that maps each edge to its group, we can now formulate the problem as

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

This formulation gives rise to a distinctive polytope, termed the *if-then polytope*, because it entails the selection of one variable each from two *if* sets of variables, which in turn implies the selection of one variable from the *then* set.

*Submitted to the editors April 20, 2024.

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

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

Related Literature. The foundational work in [28], 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, 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, 32, 23]. We refer the reader to [21] 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 [29, 34, 30, 35] together with various heuristic approaches ([13, 18, 20, 37]). Applications containing this polytope stem, for example, from the fields of data mining [26] 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 [17] and Glover's method from [16]. Another frequently utilized approach is proposed in [33]. Recently, the authors of [14] 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 [24] an efficient and 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 [25]. In recent years, multiple-choice (or set-packing) structures have also been studied in more detail. Closely related to the if-then polytope, the authors of [9] investigated the special case of BQP(G) with additional multiple-choice inequalities for partitions that apply only to the X nodes of the bipartite graph. This extension was motivated by an application to a real-world pooling problem arising in tea production. In contrast, in this paper, we consider a single multiple-choice equality for all X and all Y nodes. The bipartite quadratic assignment problem [31] and the bilinear assignment problem [38] 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.

Potential applications for if-then polytopes 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. Consider a decision where one element can be selected from a set and a set of binary variables that models the realization of another uncertain variable. Assume that this uncertain variable has influence the outcome of the decision in reality, which is again modeled by a set of binary scenario variables. Then the if-then polytope is a way to model the relationship between the decision and the two realizations. 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 [8]. The underlying clique problem with multiple-choice constraints was introduced in [10] and analyzed in [8]. In [7], 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 if-then polytopes 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 [22] in the context of optimally locating facilities. Other applications include scheduling problems ([15]), airline maintenance operations ([27]) or reactionary chemistry ([36]). A comprehensive overview of the QAP is given in [1]. An overview for different model formulations can be found in [6]. 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 an if-then substructure.

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 "if" sets imply a choice in a corresponding "then" set: the if-then polytope. 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 if-then polytope 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 tight cuts. Furthermore, we show that for a chain of several if-then polytopes, the descriptions of the individual if-then polytopes already provide a complete description of the chained 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 [19], which speeds up the solution process by orders of magnitude.

Structure of the Paper. After a short definition of the if-then polytope 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 if-then polytope. Additionally, we present efficient ways to use n-block inequalities to optimize over the if-then polytope using either a precomputation routine or a separation algorithm. Preparing the comprehensive computational study of Section 6, we first analyze the chaining of multiple if-then polytopes in Section 5, that arise in the application for stochastic railway timetabling.

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. 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:

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

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

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

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

Observation 2.1. We have $\dim(P(M)) \leq \alpha + \beta + \gamma - 3$.

Note that there are cases of M for which $\dim(P(M)) < \alpha + \beta + \gamma - 3$ 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 $P(M)$, in addition to the multiple-choice constraints.

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

LEMMA 2.2. *The vertices of $P(M)$ 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 $P(M)$ is the convex hull of a set of binary points, these are precisely the vertices of $P(M)$. ∎

It can still be benefitial to study the facet description of $P(M)$ whenever there are applications in which the determined constraints are part of a larger system. In addition to its theoretical properties, the if-then polytope has important practical applications, particularly in the field of stochastic optimization. For example, it arises naturally in the study of fixed-recourse problems in stochastic linear programming, where a decision maker faces a sequence of decisions, with the later decisions depending on the outcomes of the earlier ones. The if-then polytope can be used to model the set of feasible solutions to such problems, and to derive efficient algorithms for finding optimal solutions.

3. Valid Inequalities.

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

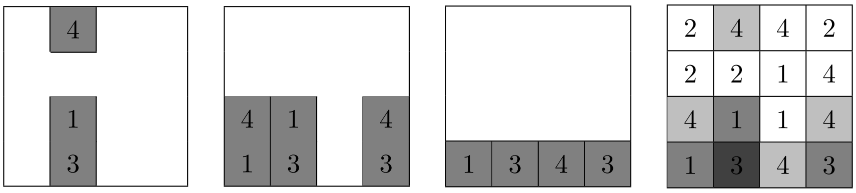


Figure 1: Construction of the 3-block inequality
$x_1 + 2x_3 + 3x_4 + 2y_1 + 3y_2 + y_3 + 2y_4 \leq 2z_1 + 3z_3 + z_4 + 3$.

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 P(M). We denote the set of z-indices contained in the block $M_{X,Y}$ by $Z^M(X,Y) := \{M_{ij} : (i,j) \in X \times Y\}$. 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*

$$\sum_{i\in X} x_i + \sum_{j\in Y} y_j \leq \sum_{l\in Z^M(X,Y)} z_l + 1, \tag{3.1}$$

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

Observation 3.1. There exist at most $(2^\alpha - 1)(2^\beta - 1)$ many non-equivalent up to scaling 1-block inequalities that are valid for $P(M)$.

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

$$\Xi^M_\kappa(K) := \{M_{ij} : |\{k \in K : (i,j) \in X_k \times Y_k\}| \geq \kappa\}. \tag{3.2}$$

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

$$\sum_{i\in[\alpha]} a_i x_i + \sum_{j\in[\beta]} b_j y_j \leq \sum_{l\in[\gamma]} c_l z_l + n, \tag{3.3}$$

where the respective variable coefficients are given by

$$a_i = |\{k \in [n] : i \in X_k\}|, \quad i \in [\alpha], \tag{3.4}$$

$$b_j = |\{k \in [n] : j \in Y_k\}|, \quad j \in [\beta], \tag{3.5}$$

$$c_l = \max\{k \in [n] : l \in \Xi^M_k([n])\}, \quad l \in [\gamma]. \tag{3.6}$$

We can use the sorting of the blocks by inclusion to efficiently determine the number of blocks intersecting in one entry (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 $k \in \{n - b_j + 1, ..., n\}$. This leads to

$$c_l = \max_{i,j\in[\alpha]\times[\beta]:M_{ij}=l} \max\{0, a_i + b_j - n\} \quad \forall l \in [\gamma]. \tag{3.7}$$

We will use this formula later in this section when characterizing n-block inequalities.

Example 3.2. Figure 1 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 colour of each entry signifies the number of blocks intersecting there. For each $l \in [4]$, the colour of the darkest cell it is contained in indicates the maximum value of κ for which l is in $\Xi_\kappa^M(\{1,2,3\})$. This value corresponds to its coefficient c_l. We can derive the colour 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 1. 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 3.3. *The n-block inequalities* (3.3) *are valid for $P(M)$ 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 (3.8a), (3.8b), the 1-block inequality (3.8c) for the intersection $M_{X_1 \cap X_2, Y_1 \cap Y_2}$ and the inequalities (3.8d) and (3.8e) derived by adding the multiple-choice constraints for the x- and y-variables and non-negativity constraints for some of the z-variables, respectively:

$$\begin{array}{llll}
(3.8a) & & \sum_{i \in X_1} x_i + \sum_{j \in Y_1} y_j - \sum_{l \in Z^M(X_1,Y_1)} z_l & \leq 1 \\
(3.8b) & + & \sum_{i \in X_2} x_i + \sum_{j \in Y_2} y_j - \sum_{l \in Z^M(X_2,Y_2)} z_l & \leq 1 \\
(3.8c) & + & \sum_{i \in X_1 \cap X_2} x_i + \sum_{j \in Y_1 \cap Y_2} y_j - \sum_{l \in Z^M(X_1 \cap X_2, Y_1 \cap Y_2)} z_l & \leq 1 \\
(3.8d) & + & \sum_{i \in [\alpha]} x_i + \sum_{j \in [\beta]} y_j & \leq 2 \\
(3.8e) & + & \sum_{l \in Z^M(X_1 \cap X_2, Y_1 \cap Y_2) \cup ((Z^M(X_1,Y_1) \cup Z^M(X_2,Y_2)) \setminus (Z^M(X_1,Y_1) \cap Z^M(X_2,Y_2)))} -z_l & \leq 0 \\
(3.8f) & = & 2 \cdot \left(\sum_{k=1}^{2} \sum_{i \in X_k} x_i + \sum_{k=1}^{2} \sum_{j \in Y_k} y_j - \sum_{k=1}^{2} \sum_{l \in \Xi_2^M([2])} z_l \right) & \leq 5 \\
(3.8g) & \overset{P(M)}{\iff} & \sum_{k=1}^{2} \sum_{i \in X_k} x_i + \sum_{k=1}^{2} \sum_{j \in Y_k} y_j - \sum_{k=1}^{2} \sum_{l \in \Xi_2^M([2])} z_l & \leq 2.
\end{array}$$

Inequality (3.8f) is valid for $P(M)$ as it is the sum of four valid inequalities. Further, all variables are binary, which implies that the 2-block inequality (3.8g) is equivalent to (3.8f) for the integer points in $P(M)$.

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

$$\sum_{i\in[\alpha]} a_i x_i + \sum_{j\in[\beta]} b_j y_j \leq \sum_{l\in[\gamma]} c_l z_l + n, \tag{3.9}$$

for $a \in \mathbb{R}^\alpha$, $b \in \mathbb{R}^\beta$ and $c \in \mathbb{R}^\gamma$ as defined in (3.4), (3.5) and (3.6), respectively. For $k \in [n]$, denote the corresponding $(n-1)$-block inequality composed of the blocks indexed by K_k as

$$\sum_{i\in[\alpha]} a_i^{K_k} x_i + \sum_{j\in[\beta]} b_j^{K_k} y_j \leq \sum_{l\in[\gamma]} c_l^{K_k} z_l + n - 1. \tag{3.10}$$

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 $n-1$, because for any $p \in \mathbb{N}$, the inequality $n(n-p)/(n-1) \geq n - p + 1$ holds iff $p \leq 1$. Therefore for $\lfloor (1/(n-1)) \sum_{k\in[n]} c_l^{K_k} \rfloor$ to be strictly greater than c_l, the inequality $c_l \geq n-1$ would have to hold. There are only two cases left to consider, namely $c_l = n$ and $c_l = n-1$. If $c_{l'} = n$ for some $l' \in [\gamma]$, then $(1/(n-1)) \sum_{k\in[n]} c_{l'}^{K_k} \leq c_{l'}$ holds, because $c_{l'}^{K_k} \leq n-1$ for all $k \in [n]$. Thus, let $c_{\tilde{l}} = n-1$ for some $\tilde{l} \in [\gamma]$. For any tuple $(i', j') \in [\alpha] \times [\beta]$ in the intersection of exactly $n-1$ blocks, there exists exactly one $k \in [n]$ for which (i', j') is not in $X_k \times Y_k$. The sorting of the blocks, i.e., $X_{k+1} \subseteq X_k$ and $Y_k \subseteq Y_{k+1}$ for all $k \in [n-1]$, implies that this one block is either the first or the last block, and since n is greater than 2, this block is not the second block. As a consequence, we obtain that $c_{\tilde{l}}^{K_2} = n-2$, and since

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

for $n \geq 3$, the relation $\lfloor (1/(n-1)) \sum_{k\in[n]} c_{\tilde{l}}^{K_k} \rfloor \leq c_{\tilde{l}}$ holds.

Now summing up all the $(n-1)$-block inequalities (3.10) 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, we obtain (3.9). ∎

3.2. Characterization of n-Block Inequalities. It is straightforward to recognize if a general inequality of the form

$$\sum_{i\in[\alpha]} a'_i x_i + \sum_{j\in[\beta]} b'_j y_j \le \sum_{l\in[\gamma]} c'_l z_l + d' \tag{3.11}$$

is an n-block inequality if $a', b', c', d' \in \mathbb{N}_0$. We first set $n := d'$. Then we construct the n blocks $M_{X,Y}$ for $k \in [n]$ via setting $X_k := \{i \in [\alpha] : a'_i \ge k\}$ and $Y_k := \{j \in [\beta] : b'_j > n - k\}$. This ensures both the sorting of the blocks by inclusion and Conditions (3.4) and (3.5). It remains to verify Condition (3.6).

Nevertheless, the addition of multiple-choice constraints for the x-, y- and z-variables and scaling may lead to inequalities that are equivalent to n-block inequalities but for which there are no subsets X_k and Y_k such that Conditions (3.4), (3.5) and (3.6) are fulfilled. We will therefore now derive three properties that any inequality of the form (3.11) possesses if it is equivalent to an n-block inequality up to addition of multiple-choice constraints and scaling. To this end, we introduce the notations $\underline{i} := \operatorname{argmin}_{i\in[\alpha]} a'_i$ and $\bar{i} := \operatorname{argmax}_{i\in[\alpha]} a'_i$, and similarly $\underline{j}$ and $\bar{j}$ as well as $\underline{l}$ and $\bar{l}$, when referring to the indices of the maximum and minimum coefficients of a', b' and c', respectively. The sorting of the blocks by inclusion implies that at least one row with index $i \in [\alpha]$ and at least one column with index $j \in [\beta]$ have to lie in each of the n blocks. Hence, in any n-block inequality, the highest occurring coefficients of the x-, y- and z-variables, respectively, are all equal to n. After adding multiple-choice constraints and scaling, this property relaxes to

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

Further, as already seen in Section 3.1,

$$c_l = \max_{i,j\in[\alpha]\times[\beta]:M_{ij}=l} \max\{0, a_i + b_j - n\} \quad \forall l \in [\gamma] \tag{3.12}$$

holds for any n-block inequality. To carry this relation between the coefficients over to n-block inequalities that have been transformed via the addition of multiple-choice constraints and scaling, we have to reverse this procedure. First, we scale the considered inequality such that all variable coefficients and the constant on the right-hand side are integer. Then we subtract adequate multiples of the three multiple-choice constraints, namely

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

such that the resulting inequality fulfills the above-mentioned property of the equality of the highest variable coefficients. As a consequence, Condition (3.12) becomes

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

Since all coefficients in an n-block inequality are non-negative, the above reverse transformation also implies the necessity of

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

The following lemma shows that Conditions (I), (II) and (III) in fact suffice to fully characterize all inequalities that are equivalent to an n-block inequality.

LEMMA 3.4. *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

$$\sum_{i\in[\alpha]} a'_i x_i + \sum_{j\in[\beta]} b'_j y_j \le \sum_{l\in[\gamma]} c'_l z_l + d', \tag{3.13}$$

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

$$\sum_{k=1}^{n} \sum_{i\in\bar{X}_k} x_i + \sum_{k=1}^{n} \sum_{j\in\bar{Y}_k} y_j \le \sum_{k=1}^{n} \sum_{l\in\Xi_k^M([n])} z_l + n, \tag{3.14}$$

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 $P(M)$:

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

Adding this equation to Inequality (3.13) yields

$$\sum_{i\in[\alpha]} (a'_i - a'_{\bar{i}} + c'_{\bar{l}} - c'_{\underline{l}})x_i + \sum_{j\in[\beta]} (b'_j - b'_{\bar{j}} + c'_{\bar{l}} - c'_{\underline{l}})y_j \leq \sum_{l\in[\gamma]} (c'_l - c'_{\underline{l}})z_l + c'_{\bar{l}} - c'_{\underline{l}}. \tag{3.15}$$

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

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

Each $i \in [\alpha]$ is contained in $(a'_i - a'_{\bar{i}} + c'_{\bar{l}} - c'_{\underline{l}})$-many sets in $\{X_k : k \in [c'_{\bar{l}} - c'_{\underline{l}}]\}$. Similarly, each $j \in [\beta]$ is contained in $(b'_j - b'_{\bar{j}} + c'_{\bar{l}} - c'_{\underline{l}})$-many sets in $\{Y_k : k \in [c'_{\bar{l}} - c'_{\underline{l}}]\}$. Further, let $n \coloneqq c'_{\bar{l}} - c'_{\underline{l}}$. Now, for (3.15) to be an n-block inequality it remains to show that each $l \in [\gamma]$ lies in the intersection of $c'_l - c'_{\underline{l}}$ and not more blocks from $\{\mathcal{X}_k \times \mathcal{Y}_k : k \in [n]\}$, i.e.,

$$\max\{k \in [c'_{\bar{l}} - c'_{\underline{l}}] : l \in \Xi^M_k([c'_{\bar{l}} - c'_{\underline{l}}])\} = c'_l - c'_{\underline{l}} \quad \forall l \in [\gamma].$$

Namely, if for any $l' \in [\gamma]$ we have $l' \in \Xi^M_k([c'_{\bar{l}} - c'_{\underline{l}}])$ for some $k > 1$, then $l' \in \Xi^M_{k-1}([c'_{\bar{l}} - c'_{\underline{l}}])$ follows trivially. The sorting of $\mathcal{X}_k$ and $\mathcal{Y}_k$ implies that for a pair $(i,j) \in [\alpha] \times [\beta]$ to be in $\mathcal{X}_{k'} \times \mathcal{Y}_{k'}$ for some $k' \in [c'_{\bar{l}} - c'_{\underline{l}}]$, the conditions $i \in \mathcal{X}_k$ for $k \in [k']$ and $j \in \mathcal{Y}_k$ for $k \in \{k', \ldots, c'_{\bar{l}} - c'_{\underline{l}}\}$ 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 entry (i,j) of M can be calculated as

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

Therefore, we need to have

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

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 $P(M)$ and are therefore useful for the branch-and-cut algorithm for solving optimization problems over $P(M)$. 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 $P(M)$. 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 4.1. *All facets of $P(M)$ are induced by either n-block inequalities or lower bounds.*

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

$$\sum_{i\in[\alpha]} a_i x_i + \sum_{j\in[\beta]} b_j y_j \leq \sum_{l\in[\gamma]} c_l z_l + d, \tag{4.1}$$

$a', b', c', d' \in \mathbb{N}_0$. Further, let $V = \{v_{t_1}, \ldots, v_{t_\nu}\}$ be a set of affine independent vertices for $\nu \coloneqq dim(P(M))$ with $V \subseteq F$. By Lemma 2.2, all vertices in V have the form

$$v_{t_k} = e_{t^x_k} + e_{\alpha+t^y_k} + e_{\alpha+\beta+M_{t^x_k t^y_k}}$$

for some $t_k = (t^x_k, t^y_k) \in [\alpha] \times [\beta]$. The tuple t_k sufficiently characterizes the vertex v_{t_k}. It indicates that $x_i = 1$ for $i = t^x_k$, $x_i = 0$ otherwise and $y_j = 1$ for $j = t^y_k$, $y_j = 0$ 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 $P(M)$, F is induced by the bound inequality $x_i \geq 0$. This holds analogously for $j \in [\beta]$ and $l \in [\gamma]$. Note that $P(M) \not\subseteq \{(x, y, z) \in \mathbb{R}^{\alpha+\beta+\gamma} : z_l = 0\}$ 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 (4.1) 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 3.4 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

$$c_l = \max_{i,j \in [\alpha] \times [\beta] : M_{ij} = l} \max\{0, a_i + b_j\} \tag{4.2}$$

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 $P(M)$ 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 $X \coloneqq \{i \in [\alpha] \setminus \{\bar{i}\} : a_i < 0\}$ and $Y \coloneqq \{j \in \beta \setminus \{\bar{j}\} : b_j < 0\}$ and suppose that X or Y is non-empty. Lifting these selected coefficients leads to a valid block inequality dominating (4.1), contradicting the assumption that (4.1) is facet-defining. Consider the inequality

$$\sum_{i \in [\alpha]} a''_i x_i + \sum_{j \in [\beta]} b''_j y_j \leq \sum_{l \in [\gamma]} c_l z_l + d', \tag{4.3}$$

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

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

and

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

and set $n' \coloneqq c_{\bar{l}}$. Now as in the proof of Lemma 3.4 the number of blocks containing the entry (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 (4.2) 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 (4.3) and the following n'-block inequality, which is valid for $P(M)$:

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

Since Inequality (4.3) dominates Inequality (4.1), the latter cannot be facet-defining, which contradicts the assumption. Thus, Condition (III) holds as well. Altogether, this means that (4.1) is equivalent to an n-block inequality. ∎

THEOREM 4.2. *The full convex-hull description of $P(M)$ is given by the multiple-choice constraints, the non-negativity constraints and the n-block constraints for $n \leq \bar{n}$ for some fixed $\bar{n} \in \mathbb{N}$.*

4.2. Separating n-Block Inequalities. To support a branch-and-cut algorithm by adding useful cuts we develope a seperation 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 4.3. *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 $\max_{i \in [\alpha]} a_i = \max_{l \in [\beta]} b_j = \max_{l \in [\gamma]} c_l = 1$ hold.*

Note that any facet 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 3.4. As a consequence, all normalized block inequalities also fulfill 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 $p = (\bar{x}, \bar{y}, \bar{z}) \notin P(M)$ with $p \geq 0$:

$$\max_{(a,b,c) \in P^{SEP}(M)} \quad \sum_{i \in [\alpha]} a_i \bar{x}_i + \sum_{j \in [\beta]} b_j \bar{y}_j - \sum_{l \in [\gamma]} c_l \bar{z}_l - 1, \tag{4.4}$$

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

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 4.4. *Assume $s \geq 0$ 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* (4.4) *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 (4.4), 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 4.5. Normalized block inequalities can be separated from points satisfying multiple-choice constraints in polynomial time.

To separate only 1-block inequalities 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 amount of z-variables in the resulting inequality is restricted to a chosen value ν.

Numerical results on our test instances in Section 6 show that the presented separation routine actually almost always separates facets if we perturb p slightly by some constant $\epsilon > 0$. But there are edge 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 4.6. 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

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

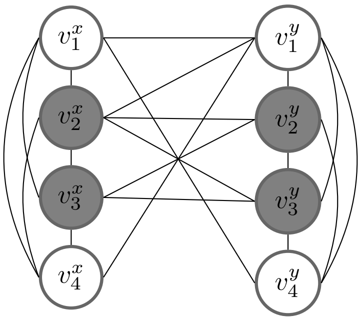
and the multiple-choice constraints

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

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 $P(M)$.

4.3. Precomputing 1-Block Inequalities Using Cliques. Experience shows that 1-block inequalities form the largest part of the facets of $P(M)$. Since it is relatively computationally easy to find good 1-block inequalities for $P(M)$ 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 $V^C(M) = V^C_X(M) \cup V^C_Y(M) = \{v^x_1, ..., v^x_\alpha\} \cup \{v^y_1, ..., v^y_\beta\}$ 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 $V^C_X(M)$ and $V^C_Y(M)$, respectively, are complete. Additionally, two nodes $v^x_i \in V^C_X(M)$ for $i \in [\alpha]$ and $v^y_j \in V^C_Y(M)$ 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 $V^C_X(M)$ times the number of selected nodes in $V^C_Y(M)$. We can either solve this clique problem exactly or use a heuristic.

Example 4.7 (Example 3.2 continued).



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

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

Figure 2 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 2. 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 If-Then. Inspired by an application in stochastic programming, we now chain several if-then-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:

$$\sum_{i \in \alpha_t} x_i^t = \sum_{j \in \beta_t} y_j^t = 1 \qquad \forall t \in [T] \tag{5.1}$$

$$x_i^t + y_{j'}^{t-1} \leq y_{M_{ij'}^t}^t + 1 \qquad \forall t \in [T], \forall i \in [\alpha_t], \forall j' \in [\beta_{t-1}]. \tag{5.2}$$

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

$$S(\mathcal{M}) := \{(x^1, \ldots, x^T, y^1, \ldots, y^T) \in \{0,1\}^{\sum_{t \in [T]} \alpha_t + \beta_t} \quad \forall t \in [T] : (5.1), (5.2)\},$$

we denote the binary feasible points for Constraints (5.1) and (5.2). We then call the convex hull of these feasible points $P(\mathcal{M}) := \operatorname{conv}(S(\mathcal{M}))$.

Lemma 5.1. *There are $\prod_{t \in T} \alpha_t$ vertices of $P(\mathcal{M})$.*

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_j^t can be derived recursively via $y_j^t = \sum_{(i,j') : M_{ij'}^{t-1} = j} x_i^t y_{j'}^{t-1}$. 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 $P(\mathcal{M})$ is the convex hull of a set of binary points, these points are all vertices of $P(\mathcal{M})$. ∎

To derive a full outer description of $P(\mathcal{M})$ 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 [8]. 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_i^t, $i \in [\alpha_t]$ and y_j^t, $j \in [\beta_t]$, is represented by a node $v_{x_i^t}$ or $v_{y_j^t}$ in $V^{\mathcal{M}}$, respectively. For each entry M_{ij}^t in the implication matrices M^t, $t \in [T]$, we further introduce a node $v_{m_{ij}^t}$. Each node is assigned to exactly one node subset, namely $v_{x_i^t}$ to V_{x^t}, $v_{y_j^t}$ to V_{y^t} and $v_{m_{ij'}^t}$ to V_{m^t}, $t \in [T]$, $i \in [\alpha_t]$, $j \in [\beta_t]$, $j' \in [\beta_{t-1}]$. Additionally, we introduce the node subset V_{y^0} containing only one node $v_{y_{j'}^0}$, where $y_{j'}^0 = 1$. 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 5.2. An integer point in $P(\mathcal{M})$ corresponds to an integer point in $P^{\text{CPMC}}(G^{\mathcal{M}}, \mathcal{V}^{\mathcal{M}})$.

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 3.

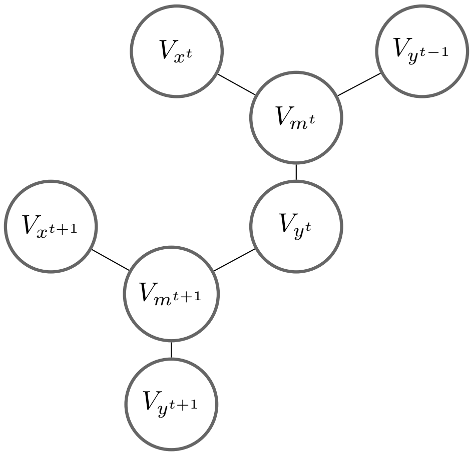


Figure 3: Dependency graph for the CPMC extension of $P(\mathcal{M})$.

It can be observed that $\mathcal{G}$ is a forest, which is the prerequisite for the main result of [8] giving a complete description for $P^{\text{CPMC}}(G, \mathcal{V})$.

THEOREM 5.3 ([8], Theorem 3.1). *Let $\mathcal{I} = (G, \mathcal{V})$ be an instance of (CPMC) with a cycle-free dependency graph. Then $P^{CPMC}(G, \mathcal{V})$ is completely described by the constraints*

$$\sum_{v \in U} x_v = 1 \quad \forall U \in \mathcal{V} \tag{5.3a}$$

$$\sum_{v \in S} x_v \leq 1 \quad \forall \textit{ stable sets } S \subseteq V \tag{5.3b}$$

$$x_v \geq 0 \quad \forall v \in V. \tag{5.3c}$$

Theorem 5.3 implies that the convex hull of the feasible points in the extended formulation of the chained if-then problem is given by the multiple-choice constraints (5.3a) on the variable sets V_{x^t}, V_{y^t} and V_{m^t} for all $t \in [T]$, the stable-set constraints (5.3b) and the non-negativity constraints (5.3c) for all variables.

Note that the nodes in the intersection of two of the subgraphs $G^{\mathcal{M}}_1, \ldots, G^{\mathcal{M}}_T$ form a stable set in $G^{\mathcal{M}}$. Since the stable-set polytope for $G^{\mathcal{M}}$ is identical to the clique polytope for its complement graph $\bar{G}^{\mathcal{M}}$, we can use the following result from [12] to state that the outer description for $P^{\text{CPMC}}(G^{\mathcal{M}}, \mathcal{V}^{\mathcal{M}})$ 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 5.4 ([12], 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 $A_1 x_1 \leq b_1$, $A_2 x_2 \leq b_2$ 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 $P(\mathcal{M})$, we use Fourier-Motzkin elimination to project the variables $m^t_{ij'}$, $i \in [\alpha_t]$, $j' \in [\beta_{t-1}]$, $t \in [T]$, out of the linear system describing the convex hull of $P^{\text{CPMC}}(G^{\mathcal{M}}, \mathcal{V}^{\mathcal{M}})$.

Each variable $m^t_{ij'}$, $i \in [\alpha_t]$, $j' \in [\beta_{t-1}]$, $t \in [T]$ is included in the inequality system describing the convex hull of $P^{\mathrm{CPMC}}(G^{\{M^t\}}, \mathcal{V}^{\{M^t\}})$ 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 $P(\mathcal{M})$ decomposes into the inequalities describing the convex hull of $P(M)$ for all $M \in \mathcal{M}$.

COROLLARY 5.5. *The polytope $P(\mathcal{M})$ is completely described by the non-negativity constraints and all n-block inequalities which are facet-defining for any of the polytopes $P(M)$, $M \in \mathcal{M}$.*

6. Computational Results. We conduct some numerical experiments to evaluate the impact of n-block inequalities on the solution time for problems which include if-then structures. We test the clique-based algorithm to precompute 1-block inequalities as described in Section 4.3, a purely cut-based solution algorithm on if-then instances with randomly generated relation matrices and the separation algorithm from Section 4.2, and customized precomputed n-block inequalities on real-world stochastic timetabling instances.

All algorithms were implemented in Python 3.10.13 using Gurobi 11.0.0 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. 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.

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. Each run involves optimizing a cost function over $P(M)$. 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 $P(M)$ equals zero.

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 $P(M)$, assuming that the facets are evenly distributed across the subsets of z-indices they contain. To evaluate this distribution, we measure the closure 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 closure as the difference between the optimal integer solution value and the optimal value of the relaxed problem with and without the precomputed 1-block cuts. We discard any run where the linear program (LP) solution equals the solution to the integer program (IP). In each run, we alternate between optimizing the LP relaxation of $P(M)$ and cutting off the resulting non-integer point using an 1-block inequality containing a fixed amount of z-variables found by the adjusted separation algorithm described in Section 4.2. Table 1 shows that increasing the

Table 1 Integrality gap closure by 1-block inequalities.

Config	$\lvert Z^M\rvert \le 1$	$\lvert Z^M\rvert \le 2$	$\lvert Z^M\rvert \le 3$	$\lvert Z^M\rvert \le 4$	$\lvert Z^M\rvert \le 5$
$\alpha = \beta = 10, \gamma = 12$	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%

number of combinations of 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 closure of the integrality gap is relativized by the number of cuts which were produced. In Table 2, the

Table 2 Integrality gap closure by 1-block inequalities per cut.

Config	$\lvert Z^M\rvert \le 1$	$\lvert Z^M\rvert \le 2$	$\lvert Z^M\rvert \le 3$	$\lvert Z^M\rvert \le 4$	$\lvert Z^M\rvert \le 5$
$\alpha=\beta=10, \gamma=12$	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%

cells of Table 1 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 closure 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

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

to the description of $P(M)$. We present the achieved integrality gap closures in Table 3. For small

Table 3 Percentage of gap closure via clique block generation.

Config	$\gamma/(\alpha\cdot\beta) = 0.0625$	$\gamma/(\alpha\cdot\beta) = 0.125$	$\gamma/(\alpha\cdot\beta) = 0.1875$	$\gamma/(\alpha\cdot\beta) = 0.25$
$\alpha=\beta=8$	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%

instances $\alpha=\beta=8$, $\gamma=4$ the integrality gap is getting closed by almost two thirds. But the amount of gap closure 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.

Table 4 Average size of the maximum blocks.

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

Nevertheless, adding 1-block inequalities computed by the presented clique-based algorithm to the description of $P(M)$ 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 $P(M)$, we can use a purely n-block-cut based solution algorithm to optimize over $P(M)$. The following test instances were generated in the same way as in the previous section. The presented measurements include the number of n-block-cuts which were used to separate non-integer solutions (Table 5) and their distribution over the number n of blocks they consist of (Table 6). As expected, 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. 1-block inequalities make up the largest part of the used cuts for

Table 5 Number of used cuts.

Config	$\gamma/(\alpha\cdot\beta)=0.04$	$\gamma/(\alpha\cdot\beta)=0.12$	$\gamma/(\alpha\cdot\beta)=0.2$	$\gamma/(\alpha\cdot\beta)=0.28$
$\alpha=\beta=5$	-	261	944	1590
$\alpha=\beta=10$	472	2511	5500	8904
$\alpha=\beta=15$	1307	6122	12551	18938
$\alpha=\beta=20$	2679	10437	20875	31465

Table 6 Distribution of n.

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%

all tested instances. Hence, even though there is no upper bound on the number of facets for if-then problems presented in this paper, the bound on the number $(2^\alpha-1)(2^\beta-1)$ of 1-block inequalities from Observation 3.1 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. One application field for if-then polytopes lies in fixed recourse stochastic programming (FRSP). The following studies are carried out on a case study for 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 [8].

The fixed recourse stochastic aspect is present in the scenario extension of the timetabling model. This feature is described in [7] and provides a way to deal with 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 [7]. 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:

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

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 if-then 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 4.

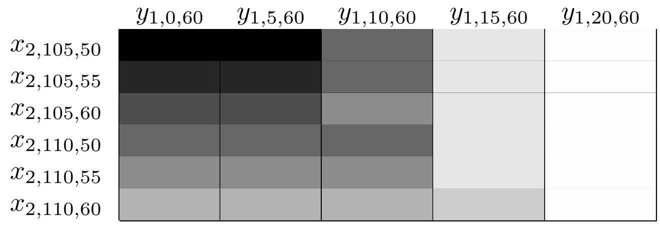


Figure 4: Example relation matrix structure for one leg and one scenario. Equal indices marked by the same gray tone.

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 for 60 instances of timetables grouped into 6 instance configurations. All instances are generated on 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 $dt|ss|nt|sn$, where for each leg in the timetable model, dt is the maximum time the departure time can be delayed or advanced, ss is the step size in the resulting time interval, nt is the number of possible running times. The number of included scenarios is given by sn. For each instance configuration, we test 10 different time horizons throughout the day, with each time horizon having a duration of 30 minutes. In order to obtain small instances which can be solved to optimality, we only optimize over one line of the train system.

Computational Results. For each test instance, we compare five solution configurations. *ORI* is the model without if-then cuts. 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 if-then cuts 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 if-then substructure in the problem if the violation is greater or equal 0.1. The number of if-then cuts added to the model is presented in Table 7. It presents the mean values for each instance configuration of constraint counts and the percentage of separated cuts which constitute facets of $P(M)$. *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. The column *Facet %* indicates the percentage of the separated cuts which are facets of $P(M)$. Since in almost all cases the separated inequalities were in fact facets we only added facets to the model and neglected the separated non-facets, without a major increase in time used in the separation routine.

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%. The time limit for the solver was set to 10 hours. This was enough time to solve each instance to optimality in at least one solution configuration. Tables 8 and 9 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 if-then cuts. If for an instance the solver did not reach the demanded gap in under 10 hours

Table 7 FRSP test: Number of constraints which are added before and after the solution process starts and percentage of separated cuts which are facets of $P(M)$.

Config	Con ORI	Con PRE	Sep SEP	Sep PRE+SEP	Facet %
10|5|2|2	19450	3124	671	319	100%
12|3|1|2	27397	3568	821	365	100%
12|6|4|2	80392	6347	726	339	99%
15|3|3|2	170094	8321	1409	814	99%
18|2|1|2	108811	6929	4039	2129	100%
10|5|1|3	37217	8637	8344	4769	97%

Table 8 FRSP test: Geometric mean runtime solving to optimality.

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

it was counted as 10 hours. The number of instances which could be solved to optimality is presented in Table 10 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 if-then cuts. 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 due to the special block structure in the relation matrix. These cuts 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. The computational study suggests, that the convex hull of if-then instances with a relation matrix structured as in this test consists of lower bounds and 1-block inequalities only. Combining the preprocessing and the separation routine we observe a significant impact of if-then cuts 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 if-then cuts 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 if-then cuts 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 if-then cuts were added.

6.3. Application to the Quadratic Assignment Problem. Koopmans and Beckmann presented a quadratic integer formulation for the quadratic assignment problem in [22]. 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

Table 9 FRSP test: Geometric mean runtime solving to a MIP optimality gap of 1%.

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

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

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

costs ($B \in \mathbb{R}_+^{m \times m}$). The quadratic integer model becomes

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

We can reformulate (QAP) into an if-then polytope 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):

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

Here, we can directly observe an if-then instance with relation matrix M^{ij}, where $M_{i_2 j_1}^{ij} := f^{ij}(i_2, j_1)$ 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.

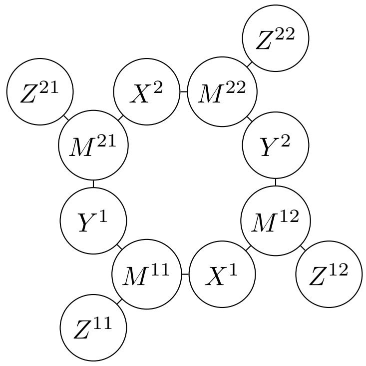


Figure 5: Dependency graph for the chaining of if-then instances of Model (QAP) with $n = 2$.

As Figure 5 shows, the dependency graph of the CPMC extension for (QAP) is not a forest, therefore n-block inequalities for the substructured if-then polytopes 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.

Instances. We analyze 28 instances from the well established QAPLIB [5]. 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 if-then 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 11.

All instances were solved with a time limit of 10 hours. Column *Sep Provided* of Table 11 shows the number of separated if-then cuts for each instance. *Sep Used* displays the number of cuts which were added to the model by Gurobi. We point out that the separation LP (4.4) exclusively produced facet-defining inequalities. The runtime to optimality or the relative MIP optimality gap in case of the time limit being exceeded for the model with (*IFTHEN*) and without (*ORIGINAL*) separated n-block inequalities are also displayed in Table 11. The shorter runtime or smaller optimality gap are marked in bold.

Results Analysis. We sorted Table 11 by the z-ratio to illustrate the strong correlation of this parameter with the positive impact of the separated if-then cuts. 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 11 indicate a positive effect of if-then cuts 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%.

7. Conclusion. In this article, we introduced the if-then polytope, a special case of the bipartite quadric polytope that models conditional relations across three sets of binary variables, where selections within two "if" sets imply a choice in a corresponding "then" 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. In a comprehensive computational study, we finally demonstrated the usefulness of n-block cuts for two

Table 11 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 (*IFTHEN*).

Config	$\gamma/(\alpha \cdot \beta)$	Sep Provided	Sep Used	ORIGINAL	IFTHEN
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%

application fields: Fixed recourse stochastic programming and the quadratic assignment problem.

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 if-then polytope. However, there is still a lot of potential for further research. On the theoretical part, the chaining of relation matrices that was present in the stochastic railway timetabling model can be extended to other tree-like structures. An increase of the number of related binary sets with multiple-choice constraints would lead to new constraint classes that can be analyzed. With regard to possible applications, we see a wide range even beyond the areas addressed so far. One promising candidate, for example, are piecewise linear relaxations for mixed-integer nonlinear programming. Here, the domain of a nonlinear function is typically divided into segments with the help of binary variables, on which a linear relaxation is then created. As only one segment can be selected, we again have a multiple choice structure. The approach in this paper can therefore be a powerful tool to tackle relationships across multiple piecewise linear relaxations of nonlinear terms.

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.

References.

[1] M. Abdel-Basset, G. Manogaran, H. Rashad, and A. N. H. Zaied, *A comprehensive review of quadratic assignment problem: variants, hybrids and applications*, Journal of Ambient Intelligence and Humanized Computing, (2018), https://doi.org/10.1007/s12652-018-0917-x.

[2] F. Barahona, M. Jünger, and G. Reinelt, *Experiments in quadratic 0–1 programming*, Mathematical programming, 44 (1989), pp. 127–137.

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

[4] E. Boros and P. L. Hammer, *Pseudo-boolean optimization*, Discrete applied mathematics, 123 (2002), pp. 155–225.

[5] R. Burkard, E. Çela, S. 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].

[6] R. E. Burkard, E. Çela, P. M. Pardalos, and L. S. Pitsoulis, *The quadratic assignment problem*, Handbook of Combinatorial Optimization, (1998), pp. 1713–1809, https://doi.org/10.1007/978-1-4613-0303-9_27, https://link.springer.com/chapter/10.1007/978-1-4613-0303-9_27.

[7] A. Bärmann, P. Gemander, and A. Martin, *A stochastic optimization approach to energy-efficient underground timetabling under uncertain dwell and running times*, Optimization Online, (2022).

[8] A. Bärmann, P. Gemander, and M. Merkert, *The clique problem with multiple-choice constraints under a cycle-free dependency graph*, Discrete Applied Mathematics, 284 (2020).

[9] A. Bärmann, A. Martin, and O. Schneider, *The bipartite boolean quadric polytope with multiple-choice constraints*, arXiv: Optimization and Control, (2020).

[10] A. Ceselli, R. Cordone, Y. Melzani, and G. 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, 2010, pp. 37–41.

[11] W.-C. Chang, S. Vakati, R. Krause, and O. Eulenstein, *Exploring biological interaction networks with tailored weighted quasi-bicliques*, in BMC bioinformatics, vol. 13, Springer, 2012, pp. 1–9.

[12] V. Chvátal, *On certain polytopes associated with graphs*, Journal of Combinatorial Theory, 18 (1975), pp. 138–154.

[13] A. Duarte, M. Laguna, R. Martí, and J. Sánchez-Oro, *Optimization procedures for the bipartite unconstrained 0-1 quadratic programming problem*, Computers & operations research, 51 (2014), pp. 123–129.

[14] R. J. Forrester and N. Hunt-Isaak, *Computational comparison of exact solution methods for 0-1 quadratic programs: Recommendations for practitioners*, Journal of Applied Mathematics, 2020 (2020), pp. 1–21.

[15] 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 (1976), pp. 595–610, https://doi.org/10.1287/OPRE.24.4.595, https://pubsonline.informs.org/doi/abs/10.1287/opre.24.4.595.

[16] F. Glover, *Improved linear integer programming formulations of nonlinear integer problems*, Management science, 22 (1975), pp. 455–460.

[17] F. Glover and E. Woolsey, *Converting the 0-1 polynomial programming problem to a 0-1 linear program*, Operations research, 22 (1974), pp. 180–182.

[18] F. Glover, T. Ye, A. P. Punnen, and G. 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 (2015), pp. 697–707.

[19] Gurobi Optimization, LLC, *Gurobi Optimizer Reference Manual*, 2023, https://www.gurobi.com.

[20] D. Karapetyan, A. P. Punnen, and A. J. Parkes, *Markov chain methods for the bipartite boolean quadratic programming problem*, European Journal of Operational Research, 260 (2017), pp. 494–506.

[21] G. Kochenberger, J.-K. Hao, F. Glover, M. Lewis, Z. Lü, H. Wang, and Y. Wang, *The unconstrained binary quadratic programming problem: a survey*, Journal of combinatorial optimization, 28 (2014), pp. 58–81.

[22] T. C. Koopmans and M. Beckmann, *Assignment problems and the location of economic activities*, Econometrica, 25 (1957), pp. 53–76, http://www.jstor.org/stable/1907742 (accessed 2024-03-20).

[23] A. N. Letchford and M. M. Sørensen, *A new separation algorithm for the boolean quadric and cut polytopes*, Discrete Optimization, 14 (2014), pp. 61–71.

[24] L. Liberti, *Compact linearization for binary quadratic problems*, 4OR, 5 (2007), pp. 231–245.

[25] R. M. Lima and I. E. Grossmann, *On the solution of nonconvex cardinality boolean quadratic programming problems: a computational study*, Computational Optimization and Applications, 66 (2017), pp. 1–37.

[26] G. Liu, K. Sim, and J. 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, Springer, 2006, pp. 437–448.

[27] A. Mason and M. Rönnqvist, *Solution methods for the balancing of jet turbines*, Computers & Operations Research, 24 (1997), pp. 153–167, https://doi.org/https://doi.org/10.1016/S0305-0548(96)00047-0, https://www.sciencedirect.com/science/article/pii/S0305054896000470.

[28] M. W. Padberg, *The boolean quadric polytope: Some characteristics, facets and relatives*, Mathematical Programming, 45 (1989), pp. 139–172.

[29] A. P. Punnen, P. Sripratak, and D. Karapetyan, *Domination analysis of algorithms for bipartite boolean quadratic programs*, in International Symposium on Fundamentals of Computation Theory, Springer, 2013, pp. 271–282.

[30] A. P. Punnen, P. Sripratak, and D. Karapetyan, *The bipartite unconstrained 0–1 quadratic programming problem: Polynomially solvable cases*, Discrete Applied Mathematics, 193 (2015), pp. 1–10.

[31] A. P. Punnen and Y. Wang, *The bipartite quadratic assignment problem and extensions*, European Journal of Operational Research, 250 (2016), pp. 715–725, https://doi.org/https://doi.org/10.1016/j.ejor.2015.10.006, https://www.sciencedirect.com/science/article/pii/S0377221715009133.

[32] H. D. Sherali, Y. Lee, and W. P. Adams, *A simultaneous lifting strategy for identifying new classes of facets for the boolean quadric polytope*, Operations Research Letters, 17 (1995), pp. 19–26.

[33] H. D. Sherali and J. C. Smith, *An improved linearization strategy for zero-one quadratic programming problems*, Optimization Letters, 1 (2007), pp. 33–47.

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

[35] P. Sripratak, A. P. Punnen, and T. Stephen, *The bipartite boolean quadric polytope*, Discrete Optimization, 44 (2022), p. 100657.

[36] I. Ugi, J. Bauer, J. Brandt, J. Friedrich, J. Gasteiger, C. Jochum, and W. Schubert, *New applications of computers in chemistry*, Angewandte Chemie International Edition in English, 18 (1979), pp. 111–123, https://doi.org/10.1002/ANIE.197901111, https://portal.fis.tum.de/en/publications/new-applications-of-computers-in-chemistry.

[37] Q. Wu, Y. Wang, and F. Glover, *Advanced tabu search algorithms for bipartite boolean quadratic programs guided by strategic oscillation and path relinking*, INFORMS Journal on Computing, 32 (2020), pp. 74–89.

[38] A. Ćustić, V. Sokol, A. P. Punnen, and B. Bhattacharya, *The bilinear assignment problem: Complexity and polynomially solvable special cases*, 2016, https://arxiv.org/abs/1605.07234.