On the Structure of Linear Programs with Overlapping Cardinality Constraints

Tobias Fischer and Marc E. Pfetsch

Department of Mathematics, TU Darmstadt, Germany
{tfischer,pfetsch}@mathematik.tu-darmstadt.de

January 25, 2017

Abstract

Cardinality constraints enforce an upper bound on the number of variables that can be nonzero. This article investigates linear programs with cardinality constraints that mutually overlap, i.e., share variables. We present the components of a branch-and-cut solution approach, including new branching rules that exploit the structure of the corresponding conflict hypergraph. We also investigate valid or facet defining cutting planes for the convex hull of the feasible solution set. Our approach can be seen as a continuous analogue of independence system polytopes. We study three different classes of cutting planes: hyperclique bound cuts, implied bound cuts, and flow cover cuts. In a computational study, we examine the effectiveness of an implementation based on the presented concepts.

1 Introduction

This article deals with *Cardinality Constrained Linear Programming Problems (CCLPs)* of the form

$$\text{(CCLP)} \qquad \begin{aligned} \max_x \quad & c^\top x \\ \text{s.t.} \quad & Ax \le b, \\ & \|x_C\|_0 \le |C| - 1 \text{ for all } C \in \mathcal{C}, \\ & x \ge 0, \end{aligned}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, for m, $n \in \mathbb{Z}_+$, and $\mathcal{C} \subseteq 2^V$ is a family of subsets of $V := \{1,\dots,n\}$. By x_C we denote the subvector of x restricted to an index set $C \subseteq V$. Moreover, $\|x_C\|_0 := |\{i \in C : x_i \neq 0\}|$ denotes the *cardinality* of x_C, i.e., the number of its nonzero entries. The sets $C \in \mathcal{C}$ are the *hyperedges* or *circuits* of the so-called *conflict hypergraph* $\mathcal{H} = (V, \mathcal{C})$, whose nodes correspond to variable indices $i \in V$. We assume that $\mathcal{H}$ is simple, i.e., $C \nsubseteq C'$ for distinct $C, C' \in \mathcal{C}$.

CCLPs form a very general class of optimization problems and consequently have a large variety of applications, e.g., in cash management (see, e.g., Galati [26]), lot sizing (see, e.g., Gade and Küçükyavuz [25]), compressed sensing (see, e.g., Sun et al. [43]), and portfolio selection (see e.g., Gao and Li [29]). This motivates the main goal of this article, which is to provide an efficient solution approach for CCLPs based on branch-and-cut. To this end, the present article investigates the two main components of branching rules and cutting planes. We will show in computational experiments that this indeed generates an effective solution method.

Not surprisingly, (CCLP) can be reformulated as a mixed integer program (MIP) if there are finite upper bounds $u \in \mathbb{R}^n_+$ on the variables. Most ideas for CCLPs are inspired by their

MIP counterparts or have to be compared to this formulation alternative. This *Mixed Integer Program with Packing Constraints* (MIPPC) formulation is:

$$
\begin{aligned}
\text{(MIPPC)} \qquad \max_{x,y} \quad & c^\top x && (1.1)\\
\text{s.t.} \quad & Ax \le b, && (1.2)\\
& \sum_{i\in C} y_i \le |C|-1 \quad \forall\, C\in\mathcal{C}, && (1.3)\\
& 0 \le x_i \le u_i\, y_i \qquad \forall\, i\in V, && (1.4)\\
& y \in \{0,1\}^n. && (1.5)
\end{aligned}
$$

Here, the cardinality requirements on the variables are enforced via packing constraints (1.3), "big-M" constraints (1.4), and auxiliary binary variables (1.5).

On the one hand, the MIPPC formulation has the advantage that it can directly be attacked using MIP-solver technology. However, it contains twice as many variables as (CCLP) and it might be the case that the x-part of a relaxation solution would be feasible for (CCLP), but the y-part still contains fractional values, which makes it infeasible for (MIPPC). The strength of its relaxation will depend on the size of the bounds u. On the other hand, (CCLP) contains fewer variables, but yields a weaker relaxation, since the cardinality constraints are not represented. Of course, this can partly be alleviated by using cutting planes, as we will show. Our computational experiments will compare the two formulations and we will discuss their respective computational benefit. It turns out that most of the techniques that we consider are more effective for overlapping cardinality constraints.

1.1 Relation to Independence Systems

The just discussed relation to the MIP formulation is highlighted by the analogue between cardinality constraints and independence systems. Given the conflict hypergraph $\mathcal{H} = (V, \mathcal{C})$, a subset $I \subseteq V$ is *independent* if I does not contain any *circuit* $C \in \mathcal{C}$. The *independence system* is given by $\mathcal{I}$, the set of all independent sets. An independent set is *maximal* or a *basis* if it is not contained in any other independent set. We denote the set of all bases by $\mathcal{B}$. Every circuit $C \in \mathcal{C}$ and all of its supersets are *dependent*. The set $\overline{\mathcal{C}} \coloneqq \{D \subseteq V \,:\, C \subseteq D,\, C \in \mathcal{C}\}$ of all dependent sets forms the so-called *dependence system* induced by $\mathcal{C}$. Note that $\overline{\mathcal{C}} = 2^V \setminus \mathcal{I}$.

The incidence vectors $x \in \{0,1\}^V$ of independent sets are characterized by the *circuit inequalities*

$$\|x_C\|_0 \le |C| - 1 \text{ for all } C \in \mathcal{C}.$$

Thus, the solutions that satisfy cardinality constraints as in (CCLP) can be seen as a continuous analogue of independence systems. Indeed, it is not hard to show that the convex hulls of the solution sets are equal, if all variables are bounded by 1, see Section 3.

Similar to the case of independence systems, we note several fundamental observations about the structure of cardinality constraints that we need throughout the article. Here, for $C \in \mathcal{C}$, we call $\|x_C\|_0 \le |C| - 1$ a *circuit constraint* and $|C|$ its order, while $\|x_K\|_0 \le k$ for $K \subseteq V$, $k \in \mathbb{Z}_+$ represents a *cardinality constraint*. Circuit constraints of order two are *complementarity constraints* equivalent to $x_i \cdot x_j = 0$ with $C = \{i, j\}$. If all circuit constraints are complementarity constraints, we speak of $\mathcal{H}$ as a conflict graph instead of a conflict hypergraph. We then denote $\mathcal{H}$ by G and its edge set by E.

We define $\mathcal{H}[K]$ with $K \subseteq V$ to be the *induced hypergraph* of $\mathcal{H}$ that has node set K and hyperedge set $\{C \in \mathcal{C} \,:\, C \subseteq K\}$. An induced hypergraph $\mathcal{H}[K]$ is called *k-hyperclique* if $\binom{K}{k} \coloneqq \{C \subseteq K \,:\, |C| = k\} \subseteq \mathcal{C}$. Due to the equivalence

$$\|x_K\|_0 \le k \quad \Longleftrightarrow \quad \|x_C\|_0 \le k \quad \forall\, C \in \tbinom{K}{k+1},$$

every cardinality constraint can be represented by circuit constraints; in this case, $\mathcal{H}[K]$ is a $(k+1)$-hyperclique. Thus, (CCLP) in fact allows to represent arbitrary cardinality constraints, which explains its name.

Cardinality constraints have a close relationship to *special ordered set type* k (SOSk) constraints, see Beale and Tomlin [6], for which the nonzero components of x_K with $\|x_K\|_0 \leq k$ have to be adjacent w.r.t. a given ordering. Note that for the special case of SOS1 constraints, the adjacency condition is void such that SOS1 constraints exactly correspond to cardinality constraints of the form $\|x_K\|_0 \leq 1$.

1.2 Literature Overview

There exist many articles that deal with the facial structure of the independence system polytope

$$P_{\text{IS}} \coloneqq \operatorname{conv}(\{\mathcal{X}(I) \in \{0,1\}^n : I \in \mathcal{I}\}),$$

where $\mathcal{X} : 2^V \to \{0,1\}^n$ is the characteristic function. This polytope or its (affinely equivalent) complementary set covering polytope was studied by, e.g., Balas and Ng [4], Sassano [39], and Sánchez-García et al. [38]. In particular, the works of Euler et al. [21] and Laurent [33] investigate cutting planes belonging to hypercliques, odd cycles, anticycles, and antiwebs for independence systems. Moreover, Easton et al. [19] and Maheshwary [34] discussed a relationship between hypercliques and facet defining inequalities of the knapsack polytope. In both of these latter works, the authors algorithmically exploit certain hypergraph structures in a branch-and-cut approach to solve particular difficult knapsack instances.

On the other hand, the polyhedral properties of cardinality constraints have been widely unexplored so far. Most studies concentrate on the special case that there is only a single cardinality constraint present in the optimization problem. For these problems, Bienstock [8] introduced classes of mixed-integer rounding inequalities, disjunctive cuts, and critical set inequalities. The latter ones were generalized by de Farias and Kozyreff [15]. In further works, de Farias et al. investigate disjunctive cuts [14] and flow cover cuts [17].

One algorithmic approach for solving (CCLP) is branch-and-cut: The reformulation of (CCLP) as (MIPPC) allows the use of state-of-the-art MIP-solvers. However, as mentioned above, big-M based modeling leads to a larger problem size and can result in numerical troubles if the upper bounds of the variables are large, see, e.g., [23] for a detailed discussion. To work around with this, Bienstock [8] was the first to introduce a specialized branching scheme, which allows a direct enforcement of the cardinality constraints without the use of auxiliary binary variables and big-M constraints. This method was taken up and generalized later by de Farias et al. [14].

Originally, the idea of specialized branching was proposed by Beale and Tomlin [6] in the early 1970s in the context of SOS1 constraints. The approach directly enforces SOS1 constraints by branching on sets of variables. In [23], we focused on the case in which the SOS1 constraints overlap and studied how to algorithmically exploit the corresponding conflict graph, for instance, to get improved branching strategies. Computational results confirm the benefit of specialized branching w.r.t. branching on an individual binary variable of (MIPPC). Moreover, de Farias et al. [16] derived classes of flow cover cuts for SOS1 constrained problems and used them in a branch-and-cut scheme; here the constraint matrix has only nonnegative coefficients. The case that also negative coefficients are present is considered in Section 8.3.

Next to branch-and-cut, there are nonlinear programming based methods known for finding local optima to optimization problems that are constrained by a single cardinality constraint. Burdakov et al. [10] introduced a reformulation as a complementarity constrained mixed-integer optimization problem for which the relaxation of the integrality conditions preserves optimal solution properties. This results in solving a mathematical program with equilibrium constraints (MPEC) for which there already exist a variety of solution approaches, see e.g., Hoheisel et al. [31].

1.3 Contribution of this Article

In Section 2, we discuss a disjunctive programming reformulation of (CCLP). This reformulation results from a representation of the circuit constraints via independent sets in the conflict hypergraph. The polyhedral properties of the feasible solution set are investigated in Section 3.

We will see in Section 4 that CCLPs are $\mathcal{NP}$-hard to solve, even for simple hypergraph classes such as single paths or matchings. However, we show in Section 5 that for particular classes, they can be solved in polynomial time. In Section 6, we investigate branching rules for (CCLP). These rules constitute a generalization of a branching scheme that we already tested computationally in [23] for the special case of complementarity constraints.

To extend our branch-and-bound algorithm to branch-and-cut, we present in Section 7 a procedure to derive valid or facet defining inequalities. In Section 8, this procedure is illustrated for three classes of cutting planes: hyperclique bound cuts, implied bound cuts, and flow cover cuts. The latter ones generalize an earlier result of de Farias et al. [16] that relies on the assumption that the constraint matrix has only nonnegative entries. Our proof covers the case of an arbitrary matrix.

In a computational study, we examine the different components of our implementation using SCIP [41, 1]. We demonstrate that for instances with a specific structure, our algorithm outperforms CPLEX 12.6.1.

2 Disjunctive Programming Formulation

Cardinality constraints have a close relationship to the theory of disjunctive programming, introduced by Balas [3]. This relationship is discussed in the following.

2.1 Linear Disjunctive Programming

Linear disjunctive programming is concerned with a finite union of linear inequality systems

$$\bigvee_{i\in C} (A^i x \leq b^i), \tag{2.1}$$

where $A^i \in \mathbb{R}^{m_i \times n}$ and $b^i \in \mathbb{R}^{m_i}$ for $i \in C$. If the polyhedra $\{x \in \mathbb{R}^n : A^i x \leq b^i\}$ do not all have the same recession cone, then (2.1) is not MIP representable, see Jeroslow [32]. However, using slack variables s_i for $i \in C$, (2.1) can be reformulated with the help of a circuit constraint as

$$\begin{aligned} &A^i x - s_i \mathbb{1} \leq b^i, \;\; s_i \geq 0 \quad \forall\, i \in C,\\ &\|s\|_0 \leq |C| - 1. \end{aligned}$$

Here, $\mathbb{1}$ denotes the vector of all ones of appropriate dimension. Thus, every linear disjunctive program can be transformed into a CCLP. Note that this reformulation can easily be generalized to nonlinear disjunctive programs. Conversely, a natural question is whether the feasible solution set of (CCLP) can be described in the form of (2.1). This is discussed in the next section.

2.2 Modeling with Disjunctive Programming

In this section, we derive a disjunctive programming reformulation of (CCLP). This reformulation results from the representation of circuit constraints via bases in the conflict hypergraph.

Let $\mathcal{I}$ denote the set of independent sets of $\mathcal{H} = (V, \mathcal{C})$ and $\mathcal{B}$ its bases, i.e., maximal independent sets. To make the transfer from the representation of (CCLP) with circuits to an equivalent representation with bases, we introduce the following notation.

Notation 2.1. Define

$$X := \{x \in \mathbb{R}^n \,:\, \|x_C\|_0 \le |C| - 1 \text{ for all } C \in \mathcal{C}\}$$

as the set of those $x \in \mathbb{R}^n$ that satisfy all the circuit constraints of (CCLP) and

$$Q := \{x \in X \,:\, Ax \le b,\, x \ge 0\}$$

as the feasible solution set of (CCLP). Moreover, for a subset $I \subseteq V$, define the sets

$$X(I) := \{x \in \mathbb{R}^n \,:\, x_i = 0 \text{ for all } i \notin I\}$$

and

$$Q(I) := \{x \in X(I) \,:\, Ax \le b,\, x \ge 0\}.$$

Based on this notation, we present four straightforward observations, which we will use throughout this section and later in the article.

Observation 2.2. *Let $\mathcal{H} = (V, \mathcal{C})$ be a hypergraph, $\mathcal{I}$ its independent sets, $\overline{\mathcal{C}} := \{D \subseteq V \,:\, C \subseteq D,\, C \in \mathcal{C}\}$ its dependent sets, and X, Q, $X(I)$, $Q(I)$ as above.*

(i) For given $I \subseteq V$ and $x \in \mathbb{R}^n$, $x \in X(I)$ if and only if $\operatorname{supp}(x) \subseteq I$.

(ii) $I \subseteq V$ is an independent set, i.e., $I \in \mathcal{I}$, if and only if $X(I) \subseteq X$.

(iii) $C \subseteq V$ is a dependent set, i.e., $C \in \overline{\mathcal{C}}$, if and only if $\|x_C\|_0 < |C|$ for all $x \in X$.

(iv) The sets $X(I)$, $I \in \mathcal{I}$, are affine subsets of X and the sets $Q(I)$, $I \in \mathcal{I}$, are polyhedral subsets of Q.

We have the following result:

Lemma 2.3. *The set X can be written as*

$$X = \bigcup_{B \in \mathcal{B}} X(B). \tag{2.2}$$

This representation is minimal in the sense that there do not exist affine spaces $\mathcal{A}_1, \dots, \mathcal{A}_k$, with $k < |\mathcal{B}|$ and $X = \mathcal{A}_1 \cup \dots \cup \mathcal{A}_k$.

Proof. Immediately from the definition, we get that $x \in X$ if and only if x satisfies

$$\bigwedge_{C \in \mathcal{C}} \bigvee_{j \in C} (x_j = 0).$$

Using the basic rules of Boolean algebra, this expression can be reformulated equivalently as

$$\bigvee_{I \in \mathcal{F}} \bigwedge_{j \in V \setminus I} (x_j = 0) \quad \iff \quad x \in \bigcup_{I \in \mathcal{F}} X(I),$$

for some $\mathcal{F} \subseteq 2^V$. In summary, this shows that $X = \bigcup_{I \in \mathcal{F}} X(I)$. Furthermore, we get that $\mathcal{F} \subseteq \mathcal{I}$ due to Observation 2.2 (ii). Consequently,

$$X = \bigcup_{I \in \mathcal{F}} X(I) \subseteq \bigcup_{I \in \mathcal{I}} X(I) = \bigcup_{B \in \mathcal{B}} X(B),$$

where the last equation results from the fact that $X(\cdot)$ is inclusion preserving. This shows the "$\subseteq$"-inclusion of (2.2).

For the converse inclusion, let $x \in \bigcup_{B \in \mathcal{B}} X(B)$. Then, for some $B^* \in \mathcal{B}$, we get that $x \in X(B^*)$, and further that $x \in X$ by Observation 2.2 *(ii)*. This completes the proof of the first assertion.

For the proof of the second assertion, we may assume without loss of generality that $|\mathcal{B}| > 2$, since in the other case X is an affine space (possibly empty). Assume that there exist affine spaces $\mathcal{A}_1, ..., \mathcal{A}_k$, with $k < |\mathcal{B}|$ and $X = \mathcal{A}_1 \cup ... \cup \mathcal{A}_k$. Let $\mathcal{X} : 2^V \to \{0,1\}^n$ be the characteristic function. Due to the fact that

$$\mathcal{X}(B) \in X = \mathcal{A}_1 \cup ... \cup \mathcal{A}_k,$$

for every $B \in \mathcal{B}$, there must exist some $j \in \{1, \dots, k\}$ such that $\mathcal{X}(B_1), \mathcal{X}(B_2) \in \mathcal{A}_j$ for at least two distinct sets $B_1, B_2 \in \mathcal{B}$. We can conclude that – as an affine space – the set $\mathcal{A}_j$ also contains $p_\lambda = (1-\lambda)\mathcal{X}(B_1) + \lambda\mathcal{X}(B_2)$ for every $\lambda \in \mathbb{R}$. Therefore, $p_\lambda \in X$ and $\operatorname{supp}(p_\lambda)$ must be an independent set. Due to $\|p_\lambda\|_0 = |B_1 \cup B_2|$ for $\lambda \in \mathbb{R} \setminus \{0,1\}$, we get a contradiction to the maximality of B_1 and B_2. $\square$

As an immediate consequence, Lemma 2.3 establishes a formulation of Q as a union of polyhedra:

Corollary 2.4. *The feasible solution set Q of (CCLP) can be covered as*

$$Q = \bigcup_{B \in \mathcal{B}} Q(B). \tag{2.3}$$

Corollary 2.5. *There exists an optimal solution of (CCLP) that is a basic feasible solution of $Q(B)$ for some $B \in \mathcal{B}$.*

We obtain the following disjunctive programming reformulation of (CCLP):

$$\text{(DP)} \qquad \max_{x \in \mathbb{R}^n} \left\{ c^\top x \, : \, \bigvee_{B \in \mathcal{B}} (x \in Q(B)) \right\}.$$

By Observation 2.2 *(i)*, the convex hull of the feasible solution set of (CCLP) can be written as the following continuous analogue to the independence system polytope:

$$\begin{aligned}
\operatorname{conv}(Q) &= \operatorname{conv}(\{x \in \mathbb{R}^n \, : \, \bigvee_{B \in \mathcal{B}} (x \in Q(B))\}) \\
&= \operatorname{conv}(\{x \in \mathbb{R}^n \, : \, Ax \le b, \, x \ge 0, \, \operatorname{supp}(x) \subseteq B, \, B \in \mathcal{B}\}) \\
&= \operatorname{conv}(\{x \in \mathbb{R}^n \, : \, Ax \le b, \, x \ge 0, \, \operatorname{supp}(x) = I, \, I \in \mathcal{I}\}).
\end{aligned}$$

This suggests an intuitive exact approach to determine an optimal solution of (CCLP). After having identified the basis set $\mathcal{B}$, it is enough to solve a number of $|\mathcal{B}|$ linear programming subproblems, each belonging to an $B \in \mathcal{B}$. Unfortunately, $|\mathcal{B}|$ may be exponential in n: For instance, a perfect matching as conflict graph, in which every node is contained in exactly one edge, has exactly $2^{n/2}$ bases. To deal with this exponential behavior, branch-and-bound may be the basis for an adequate solution strategy. We discuss several branching rules in Section 6. However, if $|\mathcal{B}|$ is small or we are only interested in an estimation of the optimal solution, explicit enumeration may be an option, see Section 5.

3 Polyhedral Properties of Sets Associated with Independence Systems

In this section, we study the polyhedral properties of $\operatorname{conv}(Q)$ and its relation to the independence system polytope P_{IS}. We will later use this to derive valid inequalities for a branch-and-cut approach.

3.1 Continuous Independence System Set

We first investigate, in which cases $\operatorname{conv}(Q)$ is a polyhedron. To give an example where this is not the case, we consider the system

$$\begin{aligned} x_1 &\geq 0, \\ 0 \leq x_2 &\leq 1, \\ x_1 \cdot x_2 &= 0. \end{aligned}$$

The set of feasible solutions is $Q = Q_1 \cup Q_2$, where $Q_1 := \{x \in \mathbb{R}^2 : x_1 \geq 0,\ x_2 = 0\}$ and $Q_2 := \{x \in \mathbb{R}^2 : x_1 = 0,\ 0 \leq x_2 \leq 1\}$. Then

$$\operatorname{conv}(Q) = \{x \in \mathbb{R}^2 : x_1 \geq 0,\ 0 \leq x_2 < 1\} \cup \{(0,1)^\top\}$$

is not a polyhedron, because it is not closed. Figure 3.1 shows a visualization.

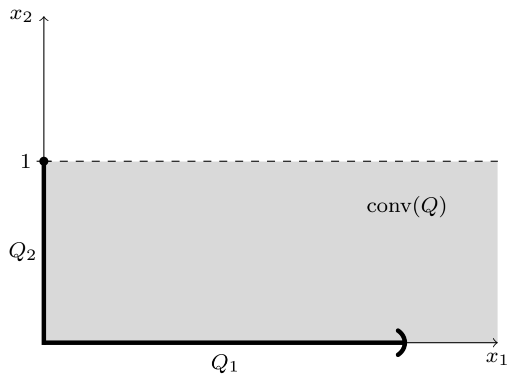

Figure 3.1: Convex hull of the union of the two polyhedra $Q_1 := \{x \in \mathbb{R}^2 : x_1 \geq 0,\ x_2 = 0\}$ and $Q_2 := \{x \in \mathbb{R}^2 : x_1 = 0, 0 \leq x_2 \leq 1\}$.

Nevertheless, the closure $\overline{\operatorname{conv}(Q)}$ of $\operatorname{conv}(Q)$ is always a polyhedron as shown by Conforti et al. [12]. To state this result, we denote $Q^0(B)$ as the vertices and $Q^\infty(B)$ as the extreme rays of $Q(B)$ for $B \in \mathcal{B}$.

Lemma 3.1 (Conforti et al. [12]). *The following complete description of $\overline{\operatorname{conv}(Q)}$ as polyhedron holds:*

$$\overline{\operatorname{conv}(Q)} = \operatorname{conv}\Big(\bigcup_{B \in \mathcal{B}} Q^0(B)\Big) + \operatorname{cone}\Big(\bigcup_{B \in \mathcal{B}} Q^\infty(B)\Big).$$

Moreover, if the sets $Q^\infty(B)$ for $B \in \mathcal{B}$ are all identical, then $\overline{\operatorname{conv}(Q)} = \operatorname{conv}(Q)$.

In order to continue the polyhedral study of $\operatorname{conv}(Q)$, we show the equivalence between (CCLP) and the MIP formulation (MIPPC). The latter is only well-defined if there exist finite upper bounds $u \in \mathbb{R}^n_+$ of x. It is worth to mention that $\overline{\operatorname{conv}(Q)} = \operatorname{conv}(Q)$ in this case, since each $Q(B)$, $B \in \mathcal{B}$, must then be a polytope with an empty extreme ray set $Q^\infty(B)$. We define S as the feasible solution set of (MIPPC), i.e., the set all those $(x, y) \in \mathbb{R}^n \times \{0,1\}^n$ with

$$Ax \leq b, \tag{3.1}$$
$$0 \leq x_i \leq u_i\, y_i \qquad \forall\, i \in V, \tag{3.2}$$
$$\sum_{i \in C} y_i \leq |C| - 1 \quad \forall\, C \in \mathcal{C}. \tag{3.3}$$

Moreover, let $S(B)$ with $B \in \mathcal{B}$ define the subset of S arising from variable fixings $y_i = 0$ and $x_i = 0$ for every $i \notin B$. By $\Pi_x(S)$, we denote the projection of S on the x-variables.

Lemma 3.2. $\Pi_x(S) = Q$.

Proof. Analogously to the proof of Lemma 2.3, one can observe that $S = \bigcup_{B\in\mathcal{B}} S(B)$. Thus, it is enough to show that $\Pi_x(S(B)) = Q(B)$ for every $B \in \mathcal{B}$: For a given $B^* \in \mathcal{B}$ let $(x, y) \in S(B^*)$. Then, by (3.2) and (3.3), x must satisfy all the circuit constraints $\|x_C\|_0 \leq |C|-1$, $C \in \mathcal{C}$, and we get that $\Pi_x(S(B^*)) \subseteq Q(B^*)$. For the converse inclusion, let $x \in Q(B^*)$. We define $y := \mathcal{X}(B^*)$ as the characteristic vector of B^*. Then $(x, y) \in S(B^*)$ and the assertion follows. □

We have seen that for the special case that $\mathrm{conv}(Q)$ is bounded, it can be completely characterized as a polyhedron. Lemma 3.2 suggests to derive valid or facet defining inequalities for $\mathrm{conv}(Q)$ from the projection of valid inequalities for $\mathrm{conv}(S)$. We will demonstrate this on the example of independence system inequalities in the next section.

3.2 Relation to Independence System Polytopes

In this section, we discuss the polyhedral relationship between (CCLP) and the independence system problem (ISP).

The convex hull of the feasible solution set of (ISP) is given by the independence system polytope $P_{\mathrm{IS}} := \mathrm{conv}(\{\mathcal{X}(I) \in \{0,1\}^n : I \in \mathcal{I}\})$, where the characteristic vectors $\mathcal{X}(I)$ of the independent sets $I \in \mathcal{I}$ represent the vertices. The polytope P_{IS} is down monotone, i.e., $z \in P_{\mathrm{IS}}$ implies $\bar{z} \in P_{\mathrm{IS}}$ for all $0 \leq \bar{z} \leq z$.

There exists an equivalent formulation of P_{IS} using circuit constraints and continuous variables. In the following, let $\mathcal{C}$ denote the minimal dependent sets (circuits) of (ISP) and X be defined according to Notation 2.1.

Lemma 3.3. $P_{\mathrm{IS}} = \mathrm{conv}(X \cap [0, \mathbb{1}])$.

Proof. Let $x \in P_{\mathrm{IS}}$. Then x can be written as a convex combination of the vertices $\mathcal{X}(I)$, $I \in \mathcal{I}$, of P_{IS}. Since by Observation 2.2 (i), $\mathcal{X}(I) \in X \cap [0, \mathbb{1}]$ for every $I \in \mathcal{I}$, it follows that $x \in \mathrm{conv}(X \cap [0, \mathbb{1}])$.

On the other hand, if $x \in \mathrm{conv}(X \cap [0, \mathbb{1}])$, then $I = \mathrm{supp}(x)$ is an independent set by Observation 2.2 (ii). Thus, $\mathcal{X}(I) \in P_{\mathrm{IS}}$. Since P_{IS} is down monotone, $x \in P_{\mathrm{IS}}$ follows. □

As an immediate consequence, we obtain the following corollaries:

Corollary 3.4. *If $Ax \leq b$ equals $x \leq \mathbb{1}$, then* $\mathrm{conv}(Q) = P_{\mathrm{IS}}$.

Corollary 3.5. *Let $0 < u_i < \infty$ for $i \in V$. The inequality $\sum_{i\in V} g_i\, y_i \leq \beta$ is valid (facet defining) for P_{IS} if and only if $\sum_{i\in V} \frac{g_i}{u_i}\, x_i \leq \beta$ is valid (facet defining) for* $\mathrm{conv}(X \cap [0, u])$.

Proof. The proof follows from Lemma 3.3 using the fact that $(x_1, \ldots, x_n) \mapsto (u_1\, x_1, \ldots, u_n\, x_n)$ defines an affine transformation mapping $\mathrm{conv}(X \cap [0, \mathbb{1}])$ to $\mathrm{conv}(X \cap [0, u])$. □

We call inequalities that are derived from this one-to-one correspondence *independence system inequalities*. One important class of independence system inequalities are *(hyperclique) bound inequalities*:

Corollary 3.6. *Let $\mathcal{H}[K]$ with $K \subseteq V$ be an induced $(k+1)$-hyperclique of $\mathcal{H} = (V, \mathcal{C})$, where $2 \leq k < |K|$. If $0 < u_i < \infty$ for $i \in K$, then the (hyperclique) bound inequality $\sum_{i\in K} \frac{x_i}{u_i} \leq k$ is valid for* $\mathrm{conv}(X \cap [0, u])$. *Moreover, if $\mathcal{H}$ is $(k+1)$-uniform, i.e., $|C| = k + 1$ for all $C \in \mathcal{C}$, and $\mathcal{H}[K]$ is maximal, i.e., $\mathcal{H}[K \cup \{v\}]$ is not a $(k+1)$-hyperclique for all $v \in V \setminus K$, then $\sum_{i\in K} \frac{x_i}{u_i} \leq k$ is facet defining for* $\mathrm{conv}(X \cap [0, u])$.

Proof. The validity of the bound inequality results from Corollary 3.5. For the proof of the facet defining property, we refer to Euler et al. [21]. □

Further independence system inequalities include representations of odd cycles, anticycles, and antiwebs (see Euler et al. [21] and Laurent [33]).

4 Complexity

In this section, we investigate complexity results for the decision problem of determining whether (CCLP) has a feasible solution. We focus on the special case that the conflict hypergraph is a standard graph, i.e., all the circuit constraints are complementarity constraints.

Most of the presently-known complexity results concerning (CCLP) deal with the *Linear Complementarity Problem (LCP)*. This fundamental problem in mathematical optimization (see, e.g., Murty [35]) is the feasibility problem of finding a vector $(x, w) \in \mathbb{R}^{2n}$, such that

$$\text{(LCP)} \qquad \begin{aligned} & w = q + Mx, \\ & x, w \geq 0, \\ & x_i \cdot w_i = 0 \qquad \forall\, i \in \{1, \ldots, n\}, \end{aligned}$$

where $q \in \mathbb{R}^n$ and $M \in \mathbb{R}^{n \times n}$. Chung [11] showed that the strongly $\mathcal{NP}$-complete problem of determining whether a linear equation system has a binary solution reduces to (LCP). His proof uses the argument that every binary constraint $y \in \{0, 1\}$ can be transformed into a complementarity constraint of the form $y\,(1-y) = 0$. In particular, this shows that the feasibility problem of (CCLP) is strongly $\mathcal{NP}$-complete for matchings as conflict graphs, i.e., graphs for which every node is contained in at most one edge. In the following, we derive complexity results for further graph classes.

Proposition 4.1. *Detecting whether (CCLP) is feasible is strongly $\mathcal{NP}$-complete, even for instances for which all circuit constraints are complementarity constraints and the conflict graph defines a cycle, or path (tree).*

Proof. We reduce the LCP as follows: For each variable pair (x_{2k-1}, x_{2k}) with $k \in \{1, \ldots, \lfloor \frac{n}{2} \rfloor\}$ and (w_{2k}, w_{2k+1}) with $k \in \{1, \ldots, \lceil \frac{n}{2} \rceil - 1\})$ of (LCP), we introduce additional variables $\bar{x}_k$ and $\bar{w}_k$, respectively. We link the new variables with the original ones by adding complementarity constraints

$$\begin{aligned} \bar{x}_k \cdot x_{2k-1} &= 0, \\ \bar{x}_k \cdot x_{2k} &= 0, \end{aligned} \qquad \text{and} \qquad \begin{aligned} \bar{w}_k \cdot w_{2k} &= 0, \\ \bar{w}_k \cdot w_{2k+1} &= 0, \end{aligned}$$

which yields, together with the complementarity constraints $x_i \cdot w_i = 0$, $i \in \{1, \ldots, n\}$, of (LCP), a path as conflict graph. Using the linear constraints $\bar{x}_k = 0$ and $\bar{w}_k = 0$, we fix the new variables to zero, and the reduction of LCP to the feasibility problem of (CCLP) with paths and trees as conflict graphs is complete. In the same way, a cycle can be established by additionally linking the pair (w_1, x_n) if n is odd or the pair (w_1, w_n) if n is even with a new variable. □

In the next section, we will prove that (CCLP) can be solved in polynomial time for co-triangulated conflict hypergraphs.

5 Polynomially Solvable Special Cases

The most frequently used type of relaxation to estimate the optimal solution of a subproblem in a branch-and-bound algorithm is the LP-relaxation. For (CCLP) this means to relax all the circuit constraints. However, we will see in this section that sometimes it is enough to relax only a couple of them in order to get a polynomially solvable problem.

Corollary 5.1. *There exists an algorithm that solves (CCLP) in polynomial time if there exists an algorithm that enumerates all bases in polynomial time in the size of (CCLP).*

Proof. The assertion follows from Corollary 2.4 and the fact that every linear problem can be solved in polynomial time in the number of variables and constraints (see Grötschel et al. [30]). □

To the best of our knowledge, there is no algorithm known that enumerates all bases $B \in \mathcal{B}$ of a general hypergraph in polynomial time in $|\mathcal{B}|$. However, there exists one for hypergraphs with bounded edge-intersections, see Boros et al. [9]. Thus, if the sizes of the edge-intersections are bounded by a constant and in addition $|\mathcal{B}|$ is polynomial in the size of (CCLP), then (CCLP) can be solved in polynomial time. In Section 6.2, we will provide a second proof of this claim.

The remainder of this section investigates a hypergraph class for which the condition of Corollary 5.1 is always satisfied. We start by introducing some notation, which is mainly taken from Berge [7] and Emtander [20]: For a given hypergraph $\mathcal{H} = (V, \mathcal{C})$ and node $v \in V$, we define

$$\Gamma(v) \coloneqq \{u \in V \,:\, u, v \in C \text{ for } u \neq v \text{ and some } C \in \mathcal{C}\}$$

as the *neighborhood* of v. Moreover, we define $\Gamma[v] \coloneqq \Gamma(v) \cup \{v\}$ as its *closed neighborhood*. A hypergraph $\mathcal{H} = (V, \mathcal{C})$ is called *d-uniform* for some $d \in \mathbb{N}$, if $|C| = d$ for every $C \in \mathcal{C}$. A d-uniform hypergraph $\mathcal{H}$ is *triangulated* if for every nonempty set $U \subseteq V$ either the induced hypergraph $\mathcal{H}[U]$ is the empty graph without hyperedges or there exists some node $v \in U$ such that the induced hypergraph $\mathcal{H}[\Gamma[v] \cap U]$ is a d-hyperclique. Triangulated graphs, as a special case, are also called chordal. The *d-complement* of a d-uniform hypergraph $\mathcal{H} = (V, \mathcal{C})$ is defined as $\mathcal{H}^c \coloneqq (V, \mathcal{C}^c)$, where $\mathcal{C}^c \coloneqq \{C \in 2^V \setminus \mathcal{C} : |C| = d\}$. We call a d-uniform hypergraph *co-triangulated* if its d-complement is triangulated.

The next result by Emtander [20] shows that there exists a polynomial-time algorithm for computing all maximal hypercliques in a d-uniform triangulated hypergraph.

Lemma 5.2 (Emtander [20])**.** *A d-uniform hypergraph $\mathcal{H} = (V, \mathcal{C})$ is triangulated if and only if it has a perfect elimination order, i.e., an ordering $v_1, \dots, v_n$ of its nodes such that for every $i \in \{1, \dots, n\}$ either the node set $K_i \coloneqq \Gamma[v_i] \cap \{v_i, \dots, v_n\}$ induces a d-hyperclique $\mathcal{H}[K_i]$ or v_i has no adjacent hyperedges in $\mathcal{H}[\{v_i, \dots, v_n\}]$.*

By definition every induced subhypergraph of a triangulated hypergraph is again triangulated. Therefore, a perfect elimination ordering in a triangulated hypergraph can be computed iteratively in polynomial time: At each iteration $i \in \{1, \dots, n\}$, search for a node v_i such that $K_i = \Gamma[v_i] \cap \{v_i, \dots, v_n\}$ induces a d-hyperclique or the empty hypergraph without hyperedges. If $\mathcal{H}[K_i]$ is not empty, then it is either a maximal hyperclique, or $K_i \subset K_j$ for some $j \in \{1, \dots, i-1\}$, provided that $i > 1$. Therefore, the number of maximal hypercliques in a d-uniform triangulated hypergraph is bounded by n. They can be enumerated inductively in this way. Since every maximal hyperclique of a d-uniform hypergraph is a basis (maximal independent set) in its d-complement hypergraph, we get the subsequent corollary:

Corollary 5.3. *Let a CCLP be given with an associated d-uniform hypergraph $\mathcal{H}$. If $\mathcal{H}$ is co-triangulated, then it has at most n bases and CCLP can be solved in polynomial time.*

Dearing et al. [18] showed that a maximal triangulated subgraph of a given graph G can be found in polynomial time. Furthermore, Dahlhaus [13] showed that a minimal triangulated graph, whose subgraph is G, can be found in polynomial time. Thus, if all circuit constraints of (CCLP) are complementarity constraints, then a valid upper bound on the objective value of (CCLP) can be computed in polynomial time by replacing its conflict graph by a maximal co-triangulated subgraph. Likewise, a lower bound can be obtained by replacing the conflict graph by a minimal co-triangulated supergraph. It remains an open question whether the results of Dearing and Dahlhaus can be generalized to triangulated hypergraphs.

6 Branching Rules

The objective of branching rules is to split the solution set of a problem into smaller parts. Standard MIP-based branching rules realize this by partitioning the domain of an individual integer variable. However, these rules are not applicable to (CCLP) whose variables are continuous. In this section, we present constraint-based branching rules that decompose the solution set by either adding new circuit or cardinality constraints to each branch or modifying the existing ones. We already tested these branching rules computationally in [23] for the special case that all circuit constraints are complementarity constraints. The purpose of this section is to generalize the results in [23] to circuit constraints and to answer open theoretical questions. Moreover, we will investigate an equivalent characterization of the constraint-based branching rules via independent sets.

Throughout this section, we consider $\mathcal{H} = (V, \mathcal{C})$ as the local conflict hypergraph, which is assumed to be simple and induced by the variables not fixed to zero. This means, in particular, that $|C| > 1$ for all $C \in \mathcal{C}$.

6.1 Structure Based Branching

We present two specialized branching rules, that enforce the circuit constraints by taking specific structures of the conflict hypergraph into account. The first one arises from the hyperclique structure and originates from work of Bienstock [8]. We give a short review of a generalized version of de Farias et al. [14]. Afterwards, we describe a second branching rule considering the neighborhood of a variable.

In the following let $\|x_K\|_0 \le k$ for $(K, k) \in \mathcal{K}$ be an initial set of cardinality constraints given for (CCLP), where the induced hypercliques $\mathcal{H}[K] = (K, \binom{K}{k+1})$ for $(K, k) \in \mathcal{K}$ together cover the hyperedges of $\mathcal{H}$; a particular example would be $\mathcal{K} = \{(C, |C| - 1) : C \in \mathcal{C}\}$.

Hyperclique branching The idea of hyperclique branching, which was introduced by de Farias et al. [14] under the term "specialized branching", is based on constraint splitting. Suppose we have given a cardinality constraint $\|x_K\|_0 \le k$, $(K, k) \in \mathcal{K}$, which is violated by the current LP solution x^*. We choose a nonempty set $S \subset K$ and a number $0 \le s < \min\{k, |S|\}$. Then a valid covering of the solution set is derived from the disjunction

$$\|x_S\|_0 \le s \qquad \vee \qquad \|x_{K \setminus S}\|_0 \le k - s - 1, \tag{6.1}$$

see [14] or Section 6.3 for the proof of correctness. The disjunction is enforced by creating two branches, where we identify $\mathcal{K} \cup \{(S, s)\}$ with the cardinality constraint set of the left branch and $\mathcal{K} \cup \{(K \setminus S, k - s - 1)\}$ with the cardinality constraint set of the right one. If $s = 0$ and $|S| = 1$, then the branching disjunction (6.1) is the one that was proposed by Bienstock [8].

Bienstock additionally suggests in [8] to use bound inequalities as a linear representation of the cardinality constraints in the LP-relaxation. This means to add $\sum_{j \in S} \frac{x_j}{u_j} \le s$ and $\sum_{j \in K \setminus S} \frac{x_j}{u_j} \le k - s - 1$ to the respective branches if finite upper bounds u_K of x_K are known.

It remains to state how to choose S and s. De Farias et al. suggest in [14] to define S as some (not further specified) set with $|S| = \lfloor |K|/2 \rfloor$ and $s = \lfloor k/2 \rfloor$. Here, we adopt a different approach that was originally proposed in the context of SOS1 constraints by Beale and Tomlin [6]: For simplicity, we assume that $K = \{1, \dots, |K|\}$. Further, we suppose that the variables x_j, $j \in K$, are sorted increasingly according to predefined weights $w \in \mathbb{R}^K$ with $0 < w_1 < \dots < w_{|K|}$. If no weights are specified beforehand, one usually takes $w_j = j$ for $j \in K$.

Our goal is to split K into $S = \{1, \dots, r\}$ and $K \setminus S = \{r + 1, \dots, |K|\}$ for some index r with $1 \le r \le |K| - 1$. We first calculate a weighted mean $\bar{w}$ as

$$\bar{w} := \frac{\sum_{j \in K} w_j \cdot x_j^*}{\sum_{j \in K} x_j^*}.$$

Notice that the denominator is nonzero by assumption. Then we choose r such that $w_r \le \bar{w} < w_{r+1}$.

Since now S is determined as $S = \{1, \dots, r\}$, it remains to select s: We suppose that all upper bounds u_K are finite. If the bound inequalities $\sum_{j \in S} \frac{x_j}{u_j} \leq s$ and $\sum_{j \in K \setminus S} \frac{x_j}{u_j} \leq k - s - 1$ are violated, then the violation of these inequalities is $\delta_1 := \sum_{j \in S} x_j^*/u_j - s$ and $\delta_2 := \sum_{j \in K \setminus S} x_j^*/u_j - k + s + 1$, respectively. To obtain balance, we choose s such that δ_1 and δ_2 are approximately equal:

$$s = \max \left\{ 0, \left\lfloor \frac{1}{2} \Big(\sum_{j \in S} \frac{x_j^*}{u_j} - \sum_{j \in K \setminus S} \frac{x_j^*}{u_j} + k - 1 \Big) \right\rfloor \right\}.$$

If the upper bounds are not finite, then we apply neighborhood branching as an alternative, see the next section.

Neighborhood branching This branching rule arises from the neighborhood of a given variable x_i, $i \in V$, in $\mathcal{H}$. The idea is to branch on the disjunction

$$x_i = 0 \qquad \vee \qquad \|x_{K \setminus \{i\}}\|_0 \leq k - 1 \quad \forall \, (K, k) \in \mathcal{K} \, : \, i \in K, \tag{6.2}$$

which is implied by $x_i = 0 \ \vee \ x_i \neq 0$. Note, however, that $x_i = 0$ is feasible for both sides of (6.2).

We implemented neighborhood branching in such a way that it does not explicitly add new cardinality constraints to each branch, i.e., the set $\mathcal{K}$ (also $\mathcal{C}$) is not enlarged during the branching process. Instead, we establish a formulation with auxiliary binary variables $y \in \{0, 1\}^V$. We use the y-variables to store the branching decisions belonging to the right hand side of (6.2), but they do not appear in the LP-relaxation. If $y_i = 1$ for some $i \in V$, then this designates that x_i can be treated as nonzero. The local cardinality constraints of each node can be reproduced from the initial cardinality constraints $\|x_K\|_0 \leq k$, $(k, K) \in \mathcal{K}$, using the model

$$\begin{aligned} &\|x_{W(K,y)}\|_0 \leq k - \sum_{j \in K} y_j, \\ &W(K, y) = K \setminus \{j \in K \, : \, y_j = 1\}, \end{aligned} \tag{6.3}$$

for $(K, k) \in \mathcal{K}$. Note that (6.3) refers to an initial cardinality constraint if no variable y_j with $j \in K$ is fixed to 1. Now (6.2) may be replaced by the equivalent and more practical disjunction $x_i = 0 \ \vee \ y_i = 1$, i.e., we fix $x_i = 0$ on the left branch and $y_i = 1$ on the right one. To improve the LP-relaxations of each branch, one can add bound inequalities $\sum_{j \in W(K,y)} \frac{x_j}{u_j} \leq k - \sum_{j \in K} y_j$ for $(K, k) \in \mathcal{K}$ if finite upper bounds $u_{W(K,y)}$ of $x_{W(K,y)}$ are known.

Example 6.1. For an illustration of neighborhood branching, consider the conflict hypergraph in Figure 6.1 whose hyperedges represent the disjunctions

$$\begin{aligned} &x_1 = 0 \ \vee \ x_5 = 0, \qquad && x_1 = 0 \ \vee \ x_2 = 0 \ \vee \ x_3 = 0 \ \vee \ x_4 = 0, \\ &x_2 = 0 \ \vee \ x_6 = 0, && x_3 = 0 \ \vee \ x_4 = 0 \ \vee \ x_5 = 0 \ \vee \ x_6 = 0. \end{aligned}$$

We apply neighborhood branching to the variable x_1, which leads to the following branching decision (according to 6.2):

$$x_1 = 0 \qquad \bigvee \qquad \begin{aligned} &x_5 = 0, \\ &x_2 = 0 \ \vee \ x_3 = 0 \ \vee \ x_4 = 0. \end{aligned} \tag{6.4}$$

An alternative branching disjunction will be demonstrated later in Example 6.4.

Observe that neighborhood branching is equivalent to standard 0/1-branching on the binary variables of (MIPPC): Branching on y_i, $i \in V$, results in $y_i = 0$ in one branch and an update of the constraints $\sum_{j \in K} y_j \leq k$ for $(K, k) \in \mathcal{K}$ with $i \in K$ to $\sum_{j \in K \setminus \{i\}} y_j \leq k - 1$ in the

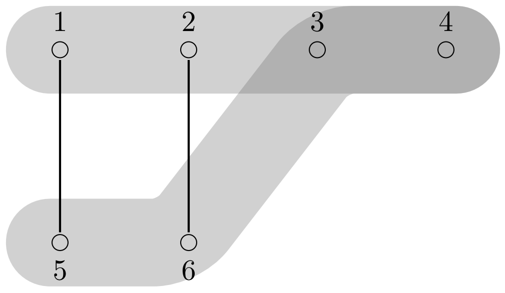

Figure 6.1: Example of a conflict hypergraph

other. Due to the big-M constraints $x_j \le u_j\, y_j$, $j \in V$, this indirectly means to branch on the disjunction (6.2).

Using neighborhood branching instead of hyperclique branching has several advantages and disadvantages: On the positive side, neighborhood branching does not explicitly add local cardinality constraints to the branching nodes, and thus keeps the memory requirements and the time needed for node-switching relatively low. Moreover, if the cardinality constraints overlap, then neighborhood branching may take multiple cardinality constraints in the enforcement of the disjunction (6.2) into account; this is not the case for hyperclique branching, which always considers only a single cardinality constraint in the branching disjunction (6.1). On the other hand, neighborhood branching can result in a very unbalanced branching tree, since it does not make the attempt to decompose the solution set evenly. Unbalanced branching can lead to relatively large enumeration; this was confirmed computationally by Appleget and Wood [2] who compared specific constraint-based branching rules like SOS branching [6] and Ryan-Foster branching [37] to the standard variable-based branching.

Implementation of the two branching rules We implemented a selection rule that decides whether to use neighborhood or hyperclique branching depending on the situation.

If all cardinality constraints $\|x_K\|_0 \le k$, $(K, k) \in \mathcal{K}$, are satisfied by the current LP solution x^*, then x^* is feasible for (CCLP). Otherwise, the selection rule chooses a pair $(K', k') \in \mathcal{K}$ whose cardinality constraint is violated by x^* and has largest value $\sum_{j \in K'} x^*_j$.

We first try to apply hyperclique branching if the depth of the current node is not larger than 20 and if all upper bounds $u_{K'}$ of $x_{K'}$ are finite. As explained above, we choose a set S and a number s from which we derive the disjunction $\|x_S\|_0 \le s \;\vee\; \|x_{K' \setminus S}\|_0 \le k' - s - 1$, see (6.1). We require that S and s should satisfy the conditions

(i) $|S| > 1$ and $|K' \setminus S| > 1$,

(ii) $\delta_1 \coloneqq \sum_{j \in S} x^*_j / u_j - s > \varepsilon$ and $\delta_2 \coloneqq \sum_{j \in K' \setminus S} x^*_j / u_j - k' + s + 1 > \varepsilon$ for some $\varepsilon \ge 0$.

If condition (i) would not be satisfied then we would fix just one variable to zero on at least one side of the branch-and-bound tree and neighborhood branching would be a better choice if the hypercliques overlap. The second condition (ii) guarantees that the bound inequalities $\sum_{j \in S} x_j / u_j \le s$ and $\sum_{j \in K' \setminus S} x_j / u_j \le k' - s - 1$ cut off x^*. Consequently, the selection rule does not always select the same cardinality constraint for branching if bound inequalities are separated with node depth frequency 1. Preliminary tests showed that a (remarkably high) value $\varepsilon = 2.0$ is an adequate choice for the instances we considered.

If the conditions for hyperclique branching are not satisfied, then we apply neighborhood branching as follows: Choose from the set K' an index i with largest value x^*_i. Then branching is performed on the disjunction $x_i = 0 \vee y_i = 1$ as explained above.

We call this combination of hyperclique and neighborhood branching *balanced branching.*

6.2 Independent Set Branching

In this section, we introduce a branch-and-bound method that we call independent set branching. Consider the feasible solution set Q of the original problem, which can be described via the union $\bigcup_{B\in\mathcal{B}} Q(B)$ over the basis set $\mathcal{B}$ (see Corollary 2.4). The key idea of independent set branching is to recursively cover Q by subsets, which themselves can be described via unions over bases. This is done by recursive partitioning of $\mathcal{B}$.

The branch-and-bound tree is constructed in the following way: The root node of the tree corresponds to the feasible solution set Q of the original problem. For branching on the root node, we select a partition $\mathcal{B}_1 \,\dot\cup\, \mathcal{B}_2$ of $\mathcal{B}$ (with $\mathcal{B}_1, \mathcal{B}_2 \neq \emptyset$). Then we create two subproblems with feasible solution set Q_r, $r \in \{1,2\}$, defined similar to Notation 2.1:

$$Q_r := \{x \in X_r \;:\; Ax \le b,\, x \ge 0\}, \qquad X_r := \bigcup_{B\in\mathcal{B}_r} X(B).$$

Recursive application of the proposed branching strategy defines a search tree that enumerates all possible solutions to (CCLP) on its leaf nodes:

Lemma 6.2. *If $\mathcal{B} = \mathcal{B}_1 \cup \mathcal{B}_2$, then $X = X_1 \cup X_2$ and $Q = Q_1 \cup Q_2$. Furthermore, if $\mathcal{B}_1 \subsetneq \mathcal{B}$ and $\mathcal{B}_2 \subsetneq \mathcal{B}$, then $X_1 \subsetneq X$ and $X_2 \subsetneq X$.*

Proof. The statements follow directly from Lemma 2.3 and Corollary 2.4. □

There exists an upper bound on the maximum number of branching nodes that the algorithm generates:

Lemma 6.3. *Let $\mathcal{B} \neq \emptyset$. The tree of subsets generated by recursive partitioning of $\mathcal{B}$ has $2\,|\mathcal{B}| - 1$ nodes.*

Proof. We use induction on $|\mathcal{B}|$: The assertion is trivial for $|\mathcal{B}| = 1$. Next, assume that $\mathcal{B}$ is partitioned into $\mathcal{B}_1 \,\dot\cup\, \mathcal{B}_2$. Let t, t_1, and t_2 be the number of nodes in the subtree associated to the sets $\mathcal{B}$, $\mathcal{B}_1$, and $\mathcal{B}_2$, respectively. Due to the induction hypothesis, we get that $t_1 = 2\,|\mathcal{B}_1| - 1$ and $t_2 = 2\,|\mathcal{B}_2| - 1$. Consequently,

$$t = t_1 + t_2 + 1 = (2\,|\mathcal{B}_1| - 1) + (2\,|\mathcal{B}_2| - 1) + 1 = 2\,|\mathcal{B}| - 1,$$

where the last equality follows from $|\mathcal{B}| = |\mathcal{B}_1| + |\mathcal{B}_2|$. □

Example 6.4. For a demonstration of independent set branching, we return to Example 6.1 and the conflict hypergraph visualized in Figure 6.1. The bases are given by

$$\begin{aligned} B_1 &= \{3,5,6\}, & B_3 &= \{2,3,4,5\}, & B_5 &= \{1,2,4\},\\ B_2 &= \{4,5,6\}, & B_4 &= \{1,2,3\}, & B_6 &= \{1,3,4,6\}. \end{aligned}$$

We apply independent set branching by using the partition of $\mathcal{B} = \{B_1, \ldots, B_6\}$ into $\mathcal{B}_1 = \{B_1, B_2, B_3\}$ and $\mathcal{B}_2 = \{B_4, B_5, B_6\}$. We obtain that

$$\begin{array}{l} x_1 = 0,\\ x_3 = 0 \;\vee\; x_4 = 0 \;\vee\; x_6 = 0, \end{array} \quad \bigvee \quad \begin{array}{l} x_5 = 0,\\ x_2 = 0 \;\vee\; x_3 = 0 \;\vee\; x_4 = 0 \end{array} \tag{6.5}$$

is a feasible disjunction, since each of the set $\{1\}$ and $\{3,4,6\}$ is not completely contained in one of the bases in $\mathcal{B}_1$ and each of the set $\{5\}$ and $\{2,3,4\}$ is not completely contained in one of the bases in $\mathcal{B}_2$. We present an efficient way to compute the disjunction (6.5) in Section 6.3.

Branching on the disjunction (6.5) results in a stronger (i.e., more tending to disjoint) coverage of the solution set than branching on the neighborhood of the variable x_1: For neighborhood branching, the condition $x_3 = 0 \;\vee\; x_4 = 0 \;\vee\; x_6 = 0$ is not added to the left hand side of the disjunction (6.4) in Example 6.1.

From Lemma 6.3, we immediately get a second proof of Corollary 5.1, which states that (CCLP) can be solved in polynomial time if there exists an algorithm that enumerates the elements of $\mathcal{B}$ in polynomial time in the problem size, as it is the case for CCLPs with co-triangulated conflict hypergraphs (see Corollary 5.3). For general conflict hypergraphs, the independent set branching rule is usually not practical, but needed to derive the branching rule of the next section.

6.3 Dependent Set Branching

Let $\mathcal{C}$ denote the circuit set of the current node and $\mathcal{B}$ the corresponding basis set. Dependent set branching is based on selecting two sets $\mathcal{C}_1, \mathcal{C}_2 \subseteq 2^V$ and to identify $\mathcal{C}_1$ with the circuit set of the left branch and $\mathcal{C}_2$ with the circuit set of the right branch. The disjunction that we enforce has the form

$$\|x_C\|_0 \leq |C| - 1 \quad \forall\, C \in \mathcal{C}_1 \setminus \mathcal{C} \quad \vee \quad \|x_C\|_0 \leq |C| - 1 \quad \forall\, C \in \mathcal{C}_2 \setminus \mathcal{C}. \tag{6.6}$$

We will investigate characterizations of $\mathcal{C}_1$ and $\mathcal{C}_2$ which result in a feasible branching rule. Moreover, we will answer the question, in which case the dependence systems induced by $\mathcal{C}_1$ and $\mathcal{C}_2$ are inclusion-wise maximal, i.e., enlarging one of the sets $\mathcal{C}_1$ or $\mathcal{C}_2$ by a circuit results in an infeasible disjunction (6.6). Note, however, that even if the dependence systems are inclusion-wise maximal, the feasible solution sets of the resulting branching nodes are not necessarily disjoint, since 0 is feasible for both if it is feasible for $Ax \leq b$. In general, a disjoint decomposition of the feasible solution set is not possible by any branching rule.

Throughout this section, we always associate $\mathcal{B}_r$ with the basis set of the hypergraph $\mathcal{H}_r := (V, \mathcal{C}_r)$, for $r \in \{1, 2\}$. Furthermore, we denote $\overline{\mathcal{C}}_r := \{D \subseteq V \,:\, C \subseteq D,\, C \in \mathcal{C}_r\}$ to be the dependence system induced by $\mathcal{C}_r$ and $\underline{\mathcal{B}}_r := \{I \subseteq V \,:\, I \subseteq B,\, B \in \mathcal{B}_r\}$ to be the independence system induced by $\mathcal{B}_r$.

One characterization of $\mathcal{C}_1$ and $\mathcal{C}_2$ that results in a feasible branching rule is the following:

Lemma 6.5. *Let two sets $\mathcal{C}_1, \mathcal{C}_2 \subseteq 2^V$ be given such that the corresponding basis sets satisfy $\mathcal{B} = \mathcal{B}_1 \cup \mathcal{B}_2$. Then (6.6) is feasible for (CCLP). If additionally $\mathcal{B}_1, \mathcal{B}_2 \subsetneq \mathcal{B}$, then $\overline{\mathcal{C}}_1, \overline{\mathcal{C}}_2 \supsetneq \overline{\mathcal{C}}$.*

Proof. Observe from Lemma 2.3 that

$$X_r := \{x \in \mathbb{R}^n \,:\, \|x_C\|_0 \leq |C| - 1 \text{ for every } C \in \mathcal{C}_r\}$$

can be written as $X_r = \bigcup_{B \in \mathcal{B}_r} X(B)$ for $r \in \{1, 2\}$. By Lemma 6.2, it holds that $X = X_1 \cup X_2$. Thus, the disjunction (6.6) is valid for X. If additionally $\mathcal{B}_1, \mathcal{B}_2 \subsetneq \mathcal{B}$, then Lemma 6.2 states that $X_1, X_2 \subsetneq X$, and consequently, $\overline{\mathcal{C}}_1, \overline{\mathcal{C}}_2 \supsetneq \overline{\mathcal{C}}$. □

It is essential for practice to have a characterization of a branching rule that is independent of basis sets and only relies on the structure of the circuit set $\mathcal{C}$. In order to get a branching rule that results in a strong (i.e., nearly disjoint) coverage of the solution set, we further need to characterize in which case $\mathcal{B} = \mathcal{B}_1 \,\dot\cup\, \mathcal{B}_2$, i.e., $\mathcal{B}_1$ and $\mathcal{B}_2$ partition $\mathcal{B}$. We need the following lemma:

Lemma 6.6. *Let two sets $\mathcal{C}_1, \mathcal{C}_2 \subseteq 2^V$ be given such that the corresponding basis sets satisfy $\mathcal{B} = \mathcal{B}_1 \,\dot\cup\, \mathcal{B}_2$. Then the following applies:*

(i) For every $C \subseteq V$, $C \in \overline{\mathcal{C}}_2$ if and only if $\{B \in \mathcal{B} \,:\, C \subseteq B\} \subseteq \mathcal{B}_1$ and $C \in \overline{\mathcal{C}}_1$ if and only if $\{B \in \mathcal{B} \,:\, C \subseteq B\} \subseteq \mathcal{B}_2$.

(ii) $\mathcal{B}_1 = \bigcup\limits_{C \in \mathcal{C}_2} \{B \in \mathcal{B} \,:\, C \subseteq B\}$ and $\mathcal{B}_2 = \bigcup\limits_{C \in \mathcal{C}_1} \{B \in \mathcal{B} \,:\, C \subseteq B\}$.

Proof. We begin with the proof of (i). Since $\overline{\mathcal{C}}_2 = 2^V \setminus \underline{\mathcal{B}}_2$, we have that

$$C \in \overline{\mathcal{C}}_2 \iff C \notin \underline{\mathcal{B}}_2 \iff \{B \in \mathcal{B}_2 \,:\, C \subseteq B\} = \emptyset \iff \{B \in \mathcal{B} \,:\, C \subseteq B\} \cap \mathcal{B}_2 = \emptyset.$$

Because $\mathcal{B} = \mathcal{B}_1 \mathbin{\dot\cup} \mathcal{B}_2$, we obtain the equivalence of $C \in \overline{\mathcal{C}}_2$ to $\{B \in \mathcal{B} \,:\, C \subseteq B\} \subseteq \mathcal{B}_1$. The second part is analogous.

To see (ii), we first observe by (i) that $\mathcal{B}_1 \supseteq \bigcup_{C \in \mathcal{C}_2} \{B \in \mathcal{B} \,:\, C \subseteq B\}$. For the reverse inclusion, let $B^* \in \mathcal{B}_1$. Then $B^* \notin \underline{\mathcal{B}}_2$, and by definition, there exists some $C \in \mathcal{C}_2$ with $C \subseteq B^*$. This proves that $B^* \in \bigcup_{C \in \mathcal{C}_2} \{B \in \mathcal{B} \,:\, C \subseteq B\}$, and hence $\mathcal{B}_1 \subseteq \bigcup_{C \in \mathcal{C}_2} \{B \in \mathcal{B} \,:\, C \subseteq B\}$. The statement for $\mathcal{B}_2$ can be shown analogously. □

Theorem 6.7. *Let two sets $\mathcal{C}_1, \mathcal{C}_2 \subseteq 2^V$ with $\overline{\mathcal{C}}_1, \overline{\mathcal{C}}_2 \supseteq \overline{\mathcal{C}}$ be given. Then the corresponding basis sets satisfy $\mathcal{B} = \mathcal{B}_1 \cup \mathcal{B}_2$ if and only if*

$$\mathcal{C}_1 \subseteq \{C_1 \subseteq V \,:\, (\forall C_2 \in \mathcal{C}_2)\ (\exists C \in \mathcal{C})\ C \subseteq C_1 \cup C_2\} =: \mathcal{G}_1, \tag{6.7}$$

$$\mathcal{C}_2 \subseteq \{C_2 \subseteq V \,:\, (\forall C_1 \in \mathcal{C}_1)\ (\exists C \in \mathcal{C})\ C \subseteq C_1 \cup C_2\} =: \mathcal{G}_2. \tag{6.8}$$

Furthermore, we have that $\mathcal{B} = \mathcal{B}_1 \mathbin{\dot\cup} \mathcal{B}_2$ if and only if $\overline{\mathcal{C}}_1 = \mathcal{G}_1$ and $\overline{\mathcal{C}}_2 = \mathcal{G}_2$ (note that $\mathcal{G}_1$ and $\mathcal{G}_2$ are dependence systems).

Proof. We start with the proof of the forward direction of the first equivalence: It is enough to show that (6.7)–(6.8) hold if $\mathcal{B} = \mathcal{B}_1 \mathbin{\dot\cup} \mathcal{B}_2$; note that $\overline{\mathcal{C}}_r = 2^V \setminus \underline{\mathcal{B}}_r$ for $r \in \{1, 2\}$ such that (6.7)–(6.8) is more difficult to meet if $\mathcal{B}_1 \cap \mathcal{B}_2 = \emptyset$. Let $C_1 \in \mathcal{C}_1$ and $C_2 \in \mathcal{C}_2$. Lemma 6.6 *(i)* implies that $\{B \in \mathcal{B} \,:\, C_1 \subseteq B\} \subseteq \mathcal{B}_2$ and $\{B \in \mathcal{B} \,:\, C_2 \subseteq B\} \subseteq \mathcal{B}_1$. Since $\mathcal{B}_1$ and $\mathcal{B}_2$ are disjoint by assumption, we therefore get

$$\{B \in \mathcal{B} \,:\, C_1 \subseteq B\} \cap \{B \in \mathcal{B} \,:\, C_2 \subseteq B\} = \emptyset.$$

Consequently, there exists no basis $B \in \mathcal{B}$ with $C_1 \cup C_2 \subseteq B$. This shows that there must exist some circuit $C \in \mathcal{C}$ with $C \subseteq C_1 \cup C_2$ and (6.7)–(6.8) follow.

We prove the backward direction of the first equivalence by contradiction: First observe that $\underline{\mathcal{B}}_1 \subseteq \underline{\mathcal{B}}$ and $\underline{\mathcal{B}}_2 \subseteq \underline{\mathcal{B}}$, since $\overline{\mathcal{C}} \subseteq \overline{\mathcal{C}}_1$ and $\overline{\mathcal{C}} \subseteq \overline{\mathcal{C}}_2$. We assume that $\underline{\mathcal{B}}_1 \cup \underline{\mathcal{B}}_2 \subsetneq \underline{\mathcal{B}}$. Then one can find some $B \in \mathcal{B}$ with $B \notin \underline{\mathcal{B}}_1, \underline{\mathcal{B}}_2$. By definition, there exist circuits $C_1 \in \mathcal{C}_1$ and $C_2 \in \mathcal{C}_2$ with $C_1 \subseteq B$ and $C_2 \subseteq B$, but there exists no set $C \in \mathcal{C}$ with $C \subseteq B$. Since $C \not\subseteq C_1 \cup C_2 \subseteq B$ for all $C \in \mathcal{C}$, we obtain that $C_1 \notin \mathcal{G}_1$ and $C_2 \notin \mathcal{G}_2$, which is a contradiction to (6.7)–(6.8). This shows that the assumption $\underline{\mathcal{B}}_1 \cup \underline{\mathcal{B}}_2 \subsetneq \underline{\mathcal{B}}$ was wrong and the claim follows.

We now proceed to the proof of the second equivalence. Define $\mathcal{B}_1^* := \bigcup_{C \in \mathcal{G}_2} \{B \in \mathcal{B} \,:\, C \subseteq B\}$ and $\mathcal{B}_2^* := \bigcup_{C \in \mathcal{G}_1} \{B \in \mathcal{B} \,:\, C \subseteq B\}$. We first show that $\mathcal{B}_1^* \cap \mathcal{B}_2^* = \emptyset$ if $\mathcal{B}_1 \mathbin{\dot\cup} \mathcal{B}_2 = \mathcal{B}$: By contradiction, suppose that $\mathcal{B}_1^* \cap \mathcal{B}_2^* \neq \emptyset$ and $\mathcal{B}_1 \mathbin{\dot\cup} \mathcal{B}_2 = \mathcal{B}$ hold. Since by (6.7)–(6.8), $\mathcal{C}_1 \subseteq \mathcal{G}_1$ and $\mathcal{C}_2 \subseteq \mathcal{G}_2$, it follows with Lemma 6.6 *(ii)* that $\mathcal{B}_1 \subseteq \mathcal{B}_1^*$ and $\mathcal{B}_2 \subseteq \mathcal{B}_2^*$. Consequently, $\mathcal{B}_1 \mathbin{\dot\cup} \mathcal{B}_2 \subseteq \mathcal{B}_1^* \cup \mathcal{B}_2^*$, and hence, $\mathcal{B} = \mathcal{B}_1 \mathbin{\dot\cup} \mathcal{B}_2 = \mathcal{B}_1^* \cup \mathcal{B}_2^* \subseteq \mathcal{B}$. Because $\mathcal{B}_1^* \cap \mathcal{B}_2^* \neq \emptyset$, we thus obtain $\mathcal{B}_1 \cap \mathcal{B}_2^* \neq \emptyset$ or $\mathcal{B}_2 \cap \mathcal{B}_1^* \neq \emptyset$. Assume without loss of generality that the first is true, and let $B \in \mathcal{B}_1 \cap \mathcal{B}_2^*$. By definition of $\mathcal{B}_2^*$, there exists some $C_1 \in \mathcal{G}_1$ with $C_1 \subseteq B$. Since (6.7) holds, it follows that for every $C_2 \in \mathcal{C}_2$ there exists some $C \in \mathcal{C}$ with $C \subseteq C_1 \cup C_2$. This implies that $C_2 \not\subseteq B$ for every $C_2 \in \mathcal{C}_2$; otherwise if $C_2 \subseteq B$, then $C \subseteq C_1 \cup C_2 \subseteq B$ and this would be a contradiction to $C \in \mathcal{C}$. By Lemma 6.6 *(ii)*, the fact that there exists some $B \in \mathcal{B}_1$ such that $C_2 \not\subseteq B$ for every $C_2 \in \mathcal{C}_2$ is a contradiction to $\mathcal{B}_1 \mathbin{\dot\cup} \mathcal{B}_2 = \mathcal{B}$. Therefore, we obtain that $\mathcal{B}_1^* \cap \mathcal{B}_2^* = \emptyset$ and consequently that $\mathcal{B}_1^* \mathbin{\dot\cup} \mathcal{B}_2^* = \mathcal{B}$ (in particular, this shows that $\mathcal{B}_1^* = \mathcal{B}_1$ and $\mathcal{B}_2^* = \mathcal{B}_2$).

We will use this finding to prove the forward direction of the second equivalence of the theorem: Supposing that $\mathcal{B}_1 \mathbin{\dot\cup} \mathcal{B}_2 = \mathcal{B}$, we obtain the inclusions $\overline{\mathcal{C}}_1 \subseteq \mathcal{G}_1$ and $\overline{\mathcal{C}}_2 \subseteq \mathcal{G}_2$ by the first equivalence of the theorem (note that $\mathcal{G}_1$ and $\mathcal{G}_2$ are dependence systems). We now show that conversely $\mathcal{G}_1 \subseteq 2^V \setminus \underline{\mathcal{B}}_1^* \subseteq \overline{\mathcal{C}}_1$. The proof that $\mathcal{G}_2 \subseteq 2^V \setminus \underline{\mathcal{B}}_2^* \subseteq \overline{\mathcal{C}}_2$ is then analogous. Let $C_1 \in \mathcal{G}_1$. By definition of $\mathcal{B}_2^*$, we have that $\{B \in \mathcal{B} : C_1 \subseteq B\} \subseteq \mathcal{B}_2^*$. Because $\mathcal{B}_1^* \mathbin{\dot\cup} \mathcal{B}_2^* = \mathcal{B}$, it follows from the backward direction in Lemma 6.6 *(i)* that $C_1 \in 2^V \setminus \underline{\mathcal{B}}_1^*$ and we obtain the inclusion $\mathcal{G}_1 \subseteq 2^V \setminus \underline{\mathcal{B}}_1^*$. The second inclusion $2^V \setminus \underline{\mathcal{B}}_1^* \subseteq \overline{\mathcal{C}}_1$ trivially follows from $\underline{\mathcal{B}}_1 = \underline{\mathcal{B}}_1^*$.

We show the backward implication of the second equivalence by contraposition: Let $\mathcal{B}_1 \cup \mathcal{B}_2 = \mathcal{B}$, but $\mathcal{B}_1 \cap \mathcal{B}_2 \neq \emptyset$. Then (6.7)–(6.8) hold by the first equivalence of the theorem. Moreover, since $\mathcal{B}_1 \subseteq \mathcal{B}_1^*$ and $\mathcal{B}_2 \subseteq \mathcal{B}_2^*$ (see above), it follows that $\mathcal{B}_1^* \cap \mathcal{B}_2^* \neq \emptyset$. Let $B \in \mathcal{B}_1^* \cap \mathcal{B}_2^*$. By definition of $\mathcal{B}_1^*$ and $\mathcal{B}_2^*$, there exists some $C_1 \in \mathcal{G}_1$ with $C_1 \subseteq B$ and some $C_2 \in \mathcal{G}_2$ with $C_2 \subseteq B$. Then $C_1 \cup C_2 \subseteq B$, and since $B \in \mathcal{B}$, there exists no circuit $C \in \mathcal{C}$ with $C \subseteq C_1 \cup C_2 \subseteq B$. By (6.7)–(6.8), it follows that $C_1 \notin \mathcal{C}_1$ and $C_2 \notin \mathcal{C}_2$. Thus, both inclusions $\overline{\mathcal{C}}_1 \subseteq \mathcal{G}_1$ and $\overline{\mathcal{C}}_2 \subseteq \mathcal{G}_2$ are proper. On the other hand, if $\overline{\mathcal{C}}_1 = \mathcal{G}_1$ and $\overline{\mathcal{C}}_2 = \mathcal{G}_2$, then $\mathcal{B}_1 \cup \mathcal{B}_2 = \mathcal{B}$ by the first equivalence of the theorem, and the negated argument of above shows that $\mathcal{B}_1 \cap \mathcal{B}_2 = \emptyset$. □

Remark 6.8. Theorem 6.7 gives a complete characterization of all circuit constraints that may be added to the branching nodes, neglecting the influence of the linear inequality system $Ax \leq b$. This is due the fact that the inclusion restrictions (6.7) and (6.8) can always be satisfied with equality for an appropriate choice of $\mathcal{C}_1$ and $\mathcal{C}_2$.

As a consequence of Theorem 6.7, we obtain that the branching disjunction of hyperclique branching (see Section 6.1) is correct.

Corollary 6.9. *Let the cardinality constraint $\|x_K\|_0 \leq k$ be valid for (CCLP). Then for every nonempty subset $S \subset K$ and number $0 \leq s < \min\{k, |S|\}$, the disjunction $\|x_S\|_0 \leq s \ \vee\ \|x_{K\setminus S}\|_0 \leq k - s - 1$ holds.*

Proof. Let $\mathcal{C}$ be the circuit set of (CCLP) and $\mathcal{B}$ the corresponding basis set. By assumption, we have that $\binom{K}{k+1} \subseteq \mathcal{C}$. Given $C_1 \in \binom{S}{s+1}$ we have for every $C_2 \in \binom{K\setminus S}{k-s}$ that $C \coloneqq C_1 \cup C_2 \in \binom{K}{k+1}$. This shows that (6.7) is satisfied by $\mathcal{C}_1 \coloneqq \binom{S}{s+1} \cup \mathcal{C}$ and $\mathcal{C}_2 \coloneqq \binom{K\setminus S}{k-s} \cup \mathcal{C}$. The proof that (6.8) holds is analogous. By Theorem 6.7, the basis sets $\mathcal{B}_1$ and $\mathcal{B}_2$ corresponding to $\mathcal{C}_1$ and $\mathcal{C}_2$ satisfy $\mathcal{B} = \mathcal{B}_1 \cup \mathcal{B}_2$ and the assertion follows from Lemma 6.5. □

Theorem 6.7 can additionally be used to complement and improve neighborhood branching: Given some node $i \in V$, neighborhood branching is performed on the disjunction

$$x_i = 0 \quad \vee \quad \|x_{C\setminus\{i\}}\|_0 \leq |C| - 2 \quad \forall\, C \in \mathcal{C} \,:\, i \in C, \tag{6.9}$$

which follows from (6.2) using $\mathcal{K} = \{(C, |C| - 1) \,:\, C \in \mathcal{C}\}$. For the new approach, we select a nonempty subset $\mathcal{M} \subseteq\coloneqq \{C \setminus \{i\} \,:\, C \in \mathcal{C},\, i \in C\}$ and define $\mathcal{C}_1, \mathcal{C}_2 \subseteq 2^V$ such that

$$\begin{aligned} \overline{\mathcal{C}}_1 &= \{C_1 \subseteq V \,:\, (\forall\, C_2 \in \mathcal{M})\ (\exists\, C \in \mathcal{C})\ C \subseteq C_1 \cup C_2\}, \\ \overline{\mathcal{C}}_2 &= \{C_2 \subseteq V \,:\, (\forall\, C_1 \in \mathcal{C}_1)\ (\exists\, C \in \mathcal{C})\ C \subseteq C_1 \cup C_2\}. \end{aligned} \tag{6.10}$$

Note that the smaller $\mathcal{M}$ is, the larger $\mathcal{C}_1$ becomes. On the other hand, the larger $\mathcal{M}$ is, the smaller $\mathcal{C}_1$ becomes and the larger $\mathcal{C}_2$ becomes.

The sets $\mathcal{C}_1$ and $\mathcal{C}_2$ have the following properties:

Corollary 6.10. *For $i \in V$, let $\mathcal{M} \subseteq \{C \setminus \{i\} \,:\, C \in \mathcal{C},\, i \in C\}$, $\mathcal{M} \neq \emptyset$, and let $\mathcal{C}_1, \mathcal{C}_2 \subseteq 2^V$ satisfy* (6.10)*. Then*

(i) $\overline{\mathcal{C}}_1 \supsetneq \overline{\mathcal{C}}$ *and* $\overline{\mathcal{C}}_2 \supsetneq \overline{\mathcal{C}}$,

(ii) $\overline{\mathcal{C}}_1 = \mathcal{G}_1$ *and* $\overline{\mathcal{C}}_2 = \mathcal{G}_2$, *and*

(iii) $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *define a feasible branching disjunction* (6.6).

Proof. Recall from the assumption stated at the beginning of Section 6 that $|C| > 1$ for all $C \in \mathcal{C}$ and that $\mathcal{H}$ is simple, i.e., $C \not\subseteq C'$ for distinct $C, C' \in \mathcal{C}$. For the proof of *(i)*, note that $\overline{\mathcal{C}}_1, \overline{\mathcal{C}}_2 \supseteq \overline{\mathcal{C}}$ by definition of $\mathcal{C}_1$ and $\mathcal{C}_2$. To show that these inclusions are strict, we observe that $\{i\} \in \overline{\mathcal{C}}_1$ by definition of $\mathcal{M}$ and that $\emptyset \neq \mathcal{M} \subseteq \overline{\mathcal{C}}_2$ by definition of $\mathcal{C}_1$. On the other hand, we have that $\{i\} \notin \overline{\mathcal{C}}$, since $|C| > 1$ for all $C \in \mathcal{C}$, and we have that $\mathcal{M} \cap \overline{\mathcal{C}} = \emptyset$, since $\mathcal{C}$ contains only inclusion-wise minimal elements.

We proceed to the proof of *(ii)*: The equality $\overline{\mathcal{C}}_2 = \mathcal{G}_2$ trivially holds by definition. To show $\overline{\mathcal{C}}_1 = \mathcal{G}_1$, we first observe that

$$\overline{\mathcal{C}}_1 \subseteq \{C_1 \subseteq V \,:\, (\forall\, C_2 \in \overline{\mathcal{C}}_2)\ (\exists\, C \in \mathcal{C})\ C \subseteq C_1 \cup C_2\} \tag{6.11}$$

by definition of $\mathcal{C}_2$. Because $\mathcal{M} \subseteq \overline{\mathcal{C}}_2$, we get the converse inclusion of (6.11) by definition of $\mathcal{C}_1$, and Statement *(ii)* is shown.

Finally, Statement *(iii)* results from Theorem 6.7 and Lemma 6.5 using *(ii)*. □

We show an application of Corollary 6.10 in the following example:

Example 6.11. Consider the conflict hypergraph visualized in Figure 6.1, whose hyperedge set is $\mathcal{C} = \{\{1,5\},\{2,6\},\{1,2,3,4\},\{3,4,5,6\}\}$. In Example 6.4 it was shown that

$$\begin{array}{l} x_1 = 0, \\ x_3 = 0 \ \vee\ x_4 = 0 \ \vee\ x_6 = 0, \end{array} \quad \bigvee \quad \begin{array}{l} x_5 = 0, \\ x_2 = 0 \ \vee\ x_3 = 0 \ \vee\ x_4 = 0 \end{array} \tag{6.12}$$

is a feasible disjunction. The two subproblems that we obtain by enforcing (6.12) have the hyperedge sets $\mathcal{C}_1 = \mathcal{C} \cup \{\{1\},\{3,4,6\}\}$ and $\mathcal{C}_2 = \mathcal{C} \cup \{\{5\},\{2,3,4\}\}$.

To derive (6.12) efficiently without computing all bases, we use Corollary 6.10: If we choose $i = 1$ and $\mathcal{M} = \{\{5\},\{2,3,4\}\}$, then for instance, $C_1 := \{3,4,6\} \in \overline{\mathcal{C}}_1$, since the union of C_1 and $C_2 = \{5\} \in \mathcal{M}$ covers $C = \{3,4,5,6\} \in \mathcal{C}$ and the union of C_1 and $C_2 = \{2,3,4\} \in \mathcal{M}$ covers $C = \{2,6\} \in \mathcal{C}$. By evaluating all possible combinations, one can show that $\overline{\mathcal{C}}_1 = \mathcal{G}_1$ and $\overline{\mathcal{C}}_2 = \mathcal{G}_2$.

We have implemented the idea of dependent set branching for the case that all circuit constraints are complementarity constraints, see [23]. Computational results confirmed that there exist choices of $\mathcal{M}$ which result in an efficient branching rule.

7 Projecting Inequalities

Recall that S denotes the feasible solution set of (MIPPC) and Q the feasible solution set of (CCLP). In this section, we derive a projection formula to transform an inequality

$$\sum_{i\in V} a_i\, x_i + \sum_{i\in V} g_i\, y_i \leq \beta$$

with $a, g \in \mathbb{R}^V$ and $\beta \in \mathbb{R}$ that is valid for $\operatorname{conv}(S)$ into an inequality that is valid for $\operatorname{conv}(Q)$. It is easy to see that if $g \geq 0$, then $\sum_{i\in V}(a_i + g_i/u_i)\, x_i \leq \beta$ is a valid inequality for $\operatorname{conv}(Q)$. Our investigations will cover the more general case that some of the g_i, $i \in V$, may be negative. Moreover, we will derive conditions for which the projected inequality defines a facet.

Throughout this section, we assume that the upper bounds $u \in \mathbb{R}^n_+$ of x are finite and nonzero. In this case, $\operatorname{conv}(Q)$ and $\operatorname{conv}(S)$ are polyhedra, see Lemma 3.1. Furthermore, we assume that $\|x_K\|_0 \leq k$ for $(K,k) \in \mathcal{K} := \{(K_r, k_r) \,:\, r \in R\}$ is an initial set of cardinality constraints given for (CCLP).

Lemma 7.1. *Let* $\sum_{i\in V} a_i\, x_i + \sum_{i\in V} g_i\, y_i \le \beta$ *be valid for* $\operatorname{conv}(S)$*. Then for every* $\lambda \in \mathbb{R}^R_+$ *satisfying* $g_i + \sum_{r\in R\,:\, i\in K_r} \lambda_r \ge 0$ *for all* $i \in V$*, the inequality*

$$\sum_{i\in V} \Big(a_i + \frac{g_i}{u_i} + \sum_{r\in R\,:\, i\in K_r} \frac{\lambda_r}{u_i} \Big)\, x_i \le \beta + \sum_{r\in R} \lambda_r\, k_r \tag{7.1}$$

is valid for $\operatorname{conv}(Q)$*.*

Proof. Assume that $\lambda \ge 0$ and $g_i + \sum_{r\in R\,:\, i\in K_r} \lambda_r \ge 0$ for all $i \in V$. Using that $x_i \le u_i\, y_i$ for $i \in V$ and $\sum_{i\in K_r} y_i \le k_r$ for $r \in R$, we obtain for $(x, y) \in \operatorname{conv}(S)$:

$$\begin{aligned} &\sum_{i\in V} \Big(a_i + \frac{g_i}{u_i} + \sum_{r\in R\,:\, i\in K_r} \frac{\lambda_r}{u_i} \Big)\, x_i - \sum_{r\in R} \lambda_r\, k_r \\ \le\; &\sum_{i\in V} a_i\, x_i + \sum_{i\in V} \Big(g_i + \sum_{r\in R\,:\, i\in K_r} \lambda_r \Big)\, y_i - \sum_{r\in R} \lambda_r\, k_r \\ =\; &\sum_{i\in V} a_i\, x_i + \sum_{i\in V} g_i\, y_i + \sum_{r\in R} \lambda_r \sum_{i\in K_r} y_i - \sum_{r\in R} \lambda_r\, k_r \\ \le\; &\sum_{i\in V} a_i\, x_i + \sum_{i\in V} g_i\, y_i \le \beta. \end{aligned} \tag{7.2}$$

This shows that (7.1) is valid for $\operatorname{conv}(S)$. The inequality is valid for $\operatorname{conv}(Q)$ as well, since it does not contain any y-variable and since $\Pi_x(S) = Q$, see Lemma 3.2. □

For Inequality (7.1) to be tight, we need to choose $\lambda \ge 0$ as small as possible, since

$$\sum_{i\in V} \sum_{r\in R\,:\, i\in K_r} \frac{\lambda_r}{u_i}\, x_i = \sum_{r\in R} \sum_{i\in K_r} \frac{\lambda_r}{u_i}\, x_i = \sum_{r\in R} \lambda_r \sum_{i\in K_r} \frac{x_i}{u_i} \le \sum_{r\in R} \lambda_r\, k_r.$$

For a special case, we can derive an explicit formula for optimally choosing λ:

Corollary 7.2. *Suppose that* $(\mathcal{H}[K_r])_{r\in R}$ *is a hyperclique partition of* V*, i.e.,* $V = \dot{\bigcup}_{r\in R} K_r$*. If* $\sum_{i\in V} a_i\, x_i + \sum_{i\in V} g_i\, y_i \le \beta$ *is valid for* $\operatorname{conv}(S)$*, then the inequality*

$$\sum_{i\in V} \Big(a_i + \frac{g_i}{u_i} - \frac{h_{r(i)}}{u_i} \Big)\, x_i \le \beta - \sum_{r\in R} h_r\, k_r \tag{7.3}$$

is valid for $\operatorname{conv}(Q)$*, where* $h_r \coloneqq \min\{\min\{0, g_j\} : j \in K_r\}$*,* $r \in R$*, and* $r(i)$*,* $i \in V$*, is the unique index* $r \in R$ *with* $i \in K_r$*.*

Proof. With $\lambda_r \coloneqq -h_r$ for $r \in R$, we obtain that $g_i + \sum_{r\in R\,:\, i\in K_r} \lambda_r = g_i - h_{r(i)}$ for $i \in V$. By definition, we have that $g_i - h_{r(i)} \ge 0$ for all $i \in V$ and $h_r \le 0$ for all $r \in R$. Hence, the assertion follows from Lemma 7.1. □

Notice that a hyperclique partition of V always exists, since the trivial constraints $\|x_i\|_0 \le 1$, $i \in V$, are valid for every instance of (CCLP). However, to obtain a strong Inequality (7.3), the constraints $\|x_{K_r}\|_0 \le k_r$, $r \in R$, should be as strong as possible.

In order to investigate facet defining properties of the developed inequalities, we define

$$\begin{aligned} F_S &\coloneqq \Big\{ (x, y) \in \operatorname{conv}(S) \;:\; \sum_{i\in V} a_i\, x_i + \sum_{i\in V} g_i\, y_i = \beta \Big\}, \\ F_Q &\coloneqq \Big\{ x \in \operatorname{conv}(Q) \;:\; \sum_{i\in V} \Big(a_i + \frac{g_i}{u_i} - \sum_{r\in R\,:\, i\in K_r} \frac{\lambda_r}{u_i} \Big)\, x_i = \beta - \sum_{r\in R} \lambda_r\, k_r \Big\}, \end{aligned}$$

where $\lambda \in \mathbb{R}^R_+$ is assumed to satisfy the conditions of Lemma 7.1. Observe by (7.2) that $x \in F_Q$ if and only if there exists some $y \in \mathbb{R}^V$ such that $(x, y) \in F_S$ and

$$\begin{aligned} x_i &= u_i\, y_i \quad \forall\, i \in L, \\ \sum_{j \in K_r} y_j &= k_r \quad \forall\, r \in T, \end{aligned} \tag{7.4}$$

where $L := \{i \in V \,:\, g_i + \sum_{r \in R\,:\, i \in K_r} \lambda_r \neq 0\}$ and $T := \{r \in R \,:\, \lambda_r \neq 0\}$. This proves the following lemma for the projection of $F'_S := \{(x, y) \in F_S \,:\, (7.4)\}$:

Lemma 7.3. *We have that* $\Pi_x(F'_S) := \{x \in \mathbb{R}^V \,:\, \exists (x, y) \in F_S \text{ with } (7.4)\} = F_Q$.

To answer the question, in which case $\Pi_x(F'_S) = F_Q$ is a facet, we will use a dimension argument. We start by repeating some general results concerning the dimension of faces of polyhedra and their projection. The following notation will be used: For a face F of a given polyhedron $P = \{(x, y) \in \mathbb{R}^p \times \mathbb{R}^q \,:\, A\, x + G\, y \leq b\}$ let $A^= x + G^= y = b^=$ denote the set of inequalities satisfied with equality for every $(x, y) \in F$. Moreover, let $\eta(F) := \operatorname{rank}(A^=, G^=)$ be the rank of $(A^=, G^=)$ and $\eta^*(F) := \operatorname{rank}(G^=)$ the rank of $G^=$. It is well-known that $\dim(F) = p + q - \eta(F)$. The dimension of the projection can be determined as follows:

Lemma 7.4 (Balas and Oosten [5]). *Let F be the face of a polyhedron $P = \{(x, y) \in \mathbb{R}^p \times \mathbb{R}^q : A\, x + G\, y \leq b\}$. Then the projection $\Pi_x(F)$ of F onto the x-variables space has dimension*

$$\dim(\Pi_x(F)) = \dim(F) - q + \eta^*(F).$$

We define $S' := \{(x, y) \in S \,:\, (7.4)\}$ and apply Lemma 7.4 to $F'_S := \{(x, y) \in F_S \,:\, (7.4)\}$ to prove the following result:

Lemma 7.5. *Let F_S be a facet of* $\operatorname{conv}(S)$. *Then $\Pi_x(F'_S) = F_Q$ is a facet of* $\operatorname{conv}(Q)$ *if and only if*

$$\eta(F'_S) - \eta(F_S) = \eta^*(\operatorname{conv}(S')) - \eta^*(\operatorname{conv}(S)) \tag{7.5}$$

holds.

Proof. For the proof of the backward direction of the theorem, assume that (7.5) holds. We show that $\Pi_x(F'_S)$ is a facet of $\operatorname{conv}(Q)$ by proving that $\dim(\Pi_x(F'_S)) = \dim(\operatorname{conv}(Q)) - 1$ as follows: By Lemma 7.4, we have that

$$\dim(\Pi_x(F'_S)) = \dim(F'_S) - n + \eta^*(F'_S). \tag{7.6}$$

Let e_i be the ith unit vector in $\mathbb{R}^n$ and $\mathbb{1}_K := \sum_{j \in K} e_j$ for $K \subseteq V$. Furthermore, let $A^= x + G^= y = b^=$ be the equality constraints of $F'_S = \{(x, y) \in F_S \,:\, (7.4)\}$. Then each vector $-u_i\, e_i$ for $i \in L$ and each vector $\mathbb{1}_{K_r}$ for $r \in T$ is a row of $G^=$. We write g as a linear combination

$$\begin{aligned} g = \sum_{i \in V} g_i\, e_i &= \sum_{i \in V} (g_i + \sum_{r \in R\,:\, i \in K_r} \lambda_r)\, e_i - \sum_{r \in R} \sum_{i \in K_r} \lambda_r\, e_i \\ &= \sum_{i \in L} (g_i + \sum_{r \in T\,:\, i \in K_r} \lambda_r)\, e_i - \sum_{r \in T} \sum_{i \in K_r} \lambda_r\, e_i = \sum_{i \in L} \frac{g_i + \sum_{r \in T\,:\, i \in K_r} \lambda_r}{u_i}\, u_i\, e_i - \sum_{r \in T} \lambda_r\, \mathbb{1}_{K_r}, \end{aligned}$$

where we use that $g_i + \sum_{r \in R\,:\, i \in K_r} \lambda_r = 0$ for all $i \in V \setminus L$ and $\lambda_r = 0$ for all $r \in R \setminus T$. We obtain that

$$\eta^*(F'_S) = \eta^*(\operatorname{conv}(S')). \tag{7.7}$$

Moreover, by the dimension formula, we have that $\dim(F'_S) = 2n - \eta(F'_S)$ and $\dim(F_S) = 2n - \eta(F_S)$, and hence:

$$\dim(F'_S) = 2n - \eta(F'_S) = \dim(F_S) - \eta(F'_S) + \eta(F_S). \tag{7.8}$$

Because $\Pi_x(S) = Q$ (see Lemma 3.2),

$$\dim(\operatorname{conv}(Q)) = \dim(\operatorname{conv}(S)) - n + \eta^*(\operatorname{conv}(S))$$

holds by Lemma 7.4. Using the fact that F_S is a facet of $\operatorname{conv}(S)$, it therefore follows that

$$\dim(F_S) = \dim(\operatorname{conv}(S)) - 1 = \dim(\operatorname{conv}(Q)) - 1 + n - \eta^*(\operatorname{conv}(S)). \tag{7.9}$$

Putting (7.6)–(7.9) and (7.5) together yields

$$\begin{aligned}\dim(\Pi(F'_S)) &\overset{(7.6)}{=} \dim(F'_S) - n + \eta^*(F'_S) \overset{(7.7)}{=} \dim(F'_S) - n + \eta^*(\operatorname{conv}(S'))\\ &\overset{(7.8)}{=} \dim(F_S) - \eta(F'_S) + \eta(F_S) - n + \eta^*(\operatorname{conv}(S'))\\ &\overset{(7.9)}{=} \dim(\operatorname{conv}(Q)) - 1 - \eta(F'_S) + \eta(F_S) - \eta^*(\operatorname{conv}(S)) + \eta^*(\operatorname{conv}(S'))\\ &\overset{(7.5)}{=} \dim(\operatorname{conv}(Q)) - 1.\end{aligned} \tag{7.10}$$

This proves the backward direction of the claim.

For the forward direction, assume that $\Pi(F'_S)$ is a facet of $\operatorname{conv}(Q)$. Then (7.5) can be derived immediately from Equation (7.10) using that $\dim(\Pi(F'_S)) = \dim(\operatorname{conv}(Q)) - 1$. $\square$

If $(\mathcal{H}[K_r])_{r\in R}$ is a hyperclique partition of V, then $L = \{i \in V : g_i \neq h_{r(i)}\}$ and $T = \{r \in R : h_r \neq 0\}$ (see the proof of Corollary 7.2). This can be used to simplify the backward statement of Lemma 7.5 for a special case:

Corollary 7.6. *Suppose that $(\mathcal{H}[K_r])_{r\in R}$ is a hyperclique partition of V and that $\operatorname{conv}(S)$ is full dimensional. If*

(i) F_S is a facet of $\operatorname{conv}(S)$,

(ii) F'_S is a facet of $\operatorname{conv}(S')$, and

(iii) $\eta(\operatorname{conv}(S')) = |L| + |T|$,

then $\Pi_x(F'_S) = F_Q$ is a facet of $\operatorname{conv}(Q)$.

Proof. By the dimension formula, we obtain from *(i)* that $\eta(\operatorname{conv}(S)) = \eta(F_S) - 1$ and from *(ii)* that $\eta(\operatorname{conv}(S')) = \eta(F'_S) - 1$. Consequently,

$$\eta(F'_S) - \eta(F_S) = \eta(\operatorname{conv}(S')) - \eta(\operatorname{conv}(S)). \tag{7.11}$$

Moreover, with *(iii)* we get

$$|L| + |T| \overset{(iii)}{=} \eta(\operatorname{conv}(S')) \geq \eta^*(\operatorname{conv}(S')) \geq \eta^*(\{(x,y) \in \mathbb{R}^{2n} : (7.4)\}). \tag{7.12}$$

Next, we show that $\eta^*(\{(x,y) \in \mathbb{R}^{2n} : (7.4)\}) = |L| + |T|$ by proving that the vectors e_i, for $i \in L$, and $\mathbb{1}_{K_r}$, for $r \in T$, are linearly independent: Since the hypercliques are nonoverlapping, we have that $K_r \cap K_{r^*} = \emptyset$, for distinct $r, r^* \in T$. Thus, the vectors $\mathbb{1}_{K_r}$ for $r \in T$ are linearly independent. Furthermore,

$$L \cap K_r = \{j \in K_r : g_j \neq h_r\} \neq K_r,$$

for all $r \in T$, where we use that $h_r = \min\{\min\{0, g_j\} : j \in K_r\} \neq 0$ by definition of $r \in T$. Therefore, every vector $\mathbb{1}_{K_r}$, for $r \in T$, cannot be written as a linear combination of vectors e_i, for $i \in L$. In summary, this shows that $\eta^*(\{(x, y) \in \mathbb{R}^{2n} : (7.4)\}) = |L| + |T|$, and Inequality (7.12) is satisfied with equality; i.e., in particular,

$$\eta(\mathrm{conv}(S')) = \eta^*(\mathrm{conv}(S')). \tag{7.13}$$

Further, since $\mathrm{conv}(S)$ is full dimensional, it follows that $\eta^*(\mathrm{conv}(S)) = \eta(\mathrm{conv}(S)) = 0$. Together with (7.11) and (7.13), this proves

$$\eta(F'_S) - \eta(F_S) = \eta^*(\mathrm{conv}(S')) - \eta^*(\mathrm{conv}(S)).$$

The assertion follows from Lemma 7.5. □

Applications for the results presented above are discussed in the next section.

8 Cutting Planes

In this section, we derive valid cutting planes for (CCLP) following the procedure of the former section and the procedure of Section 3.2. We focus on three different classes of cutting planes: hyperclique bound cuts, implied bound cuts, and flow cover cuts. The latter cuts generalize the ones presented in de Farias et al. [16] to the case in which the constraint matrix has both positive and negative entries.

8.1 Hyperclique Bound Cuts

Hyperclique bound cuts were introduced by Bienstock [8] to strengthen the LP-relaxation of (CCLP). Consider an induced $(k+1)$-hyperclique $\mathcal{H}[K]$ of $\mathcal{H}$ representing a cardinality constraint of the form $\|x_K\|_0 \leq k$. If every variable x_i, $i \in K$, has a finite upper bound $u_i \in \mathbb{R}_{>0}$, then one can add a *(hyperclique) bound inequality* of the form

$$\sum_{i \in K} \frac{x_i}{u_i} \leq k$$

to the LP-relaxation. This inequality is derived from the projection of the inequality $\sum_{i \in K} y_i \leq k$ for (MIPPC) to the x-variables space, see Corollary 3.6.

In our implementation, we separate bound inequalities from the initial cardinality constraints given for (CCLP) and the local cardinality constraints generated by hyperclique branching, see Section 6.1. If the cardinality constraints overlap, then one can obtain in principle tighter inequalities by searching for maximal hypercliques in $\mathcal{H}$. The separation problem can be formulated as a maximum weighted hyperclique problem. However, this has not yet been implemented.

8.2 Implied Bound Cuts

Savelsbergh [40] describes preprocessing and probing techniques to derive logical implications between variables for MIPs. From these relationships one can deduce implied bound cuts. This section presents an adaptation to the context of (CCLP) by considering implications between two continuous variables.

Let x and z be variables, where we assume that $x \leq u$ and $0 \leq z \leq d$ for $u, d \in \mathbb{R}_{>0}$. An implication between these variables can be of the form

$$z > 0 \implies x \leq \lambda, \tag{8.1}$$

with $\lambda \in \mathbb{R}$ and $\lambda < u$. There exists an analogous variant with an implied lower bound for x. From every implication, one can deduce an *implied bound inequality*

$$x + \frac{u - \lambda}{d} z \leq u \tag{8.2}$$

that can serve as a cutting plane in a branch-and-cut algorithm.

Inequality (8.2) is just a more direct version of implied bound inequalities for (MIPPC): With the help of an auxiliary binary variable y, (8.1) can be written as

$$\begin{aligned} y = 1 &\implies x \leq \lambda, \\ y = 0 &\implies z \leq 0. \end{aligned}$$

The associated implied bound cuts are $x + (u - \lambda)\, y \leq u$ and $z \leq d\, y$. We obtain

$$x + \frac{u - \lambda}{d} z \leq x + (u - \lambda)\, y \leq u, \tag{8.3}$$

which proves the validity of Inequality (8.2). Note that the procedure in (8.3) is analogous to the proof of the Projection Lemma 7.1.

Example 8.1. To demonstrate the effectiveness of implied bound cuts, we consider the following example:

$$\begin{aligned} \max \quad & 2x + z \\ \text{s.t.} \quad & x - w \leq \frac{1}{2}, \\ & z \cdot w = 0, \\ & 0 \leq x, z, w \leq 1. \end{aligned}$$

The optimal value of the LP-relaxation is 3. If $z > 0$, then this implies that $w = 0$ and subsequently that $x \leq 1/2$. If we add the (clique) bound inequality $z + w \leq 1$ arising from $z \cdot w = 0$ to the problem, then we obtain an optimal LP value of $5/2$. However, if we instead (or additionally) add the implied bound inequality $x + 1/2\, z \leq 1$, then the optimal LP value is 2 and coincides with the optimal value of the original problem.

8.3 Flow Cover Cuts

In this section, we restrict our attention to the case in which all circuit constraints of (CCLP) are complementarity constraints. For the respective problem class, we investigate strong valid inequalities that we derive from a projection of MIP-based flow cover cuts. These inequalities generalize cutting planes of de Farias et al. [16] to the case in which the constraint matrix has both positive and negative entries.

As usual let $V = \{1, \ldots, n\}$ and let $N^+, N^- \subseteq V$ be disjoint sets. We consider an SOS1 constrained single node network design model of the form

$$\text{(ND)} \qquad \begin{aligned} & \sum_{i \in N^+} x_i - \sum_{i \in N^-} x_i \leq \beta, \\ & 0 \leq x_i \leq u_i && \forall\, i \in V, \\ & \|x_{K_r}\|_0 \leq 1 && \forall\, r \in R, \end{aligned}$$

where $u \in (\mathbb{R}_{>0})^V$, $\beta \in \mathbb{R}$, and $K_r \subseteq V$ for $r \in R$. By P_{ND} we denote the convex hull of the feasible solution set of (ND). Moreover, given $\mu \in \mathbb{R}$ we define $\mu^+ := \max\{0, \mu\}$.

(ND) can be converted to mixed-integer form with the help of auxiliary binary variables:

$$
\text{(ND}^*\text{)} \qquad \begin{aligned}
&\sum_{i\in N^+} x_i - \sum_{i\in N^-} x_i \le \beta, && \\
&0 \le x_i \le u_i\, y_i && \forall\, i \in V, \\
&\sum_{j\in K_r} y_j \le 1 && \forall\, r \in R, \\
&y \in \{0,1\}^V. &&
\end{aligned}
$$

For simplicity, we assume that the cliques $(K_r)_{r\in R}$ partition V, i.e., $V = \dot{\bigcup}_{r\in R} K_r$. Then a pair (C^+, C^-) with $C^+ \subseteq N^+$ and $C^- \subseteq N^-$ is said to be a *flow cover* if $|K_r \cap (C^+ \cup C^-)| \le 1$ for all $r \in R$ and

$$\lambda \coloneqq \sum_{i\in C^+} u_i - \sum_{i\in C^-} u_i - \beta > 0.$$

For several special cases, the polyhedral properties for both (ND) and (ND*) have been investigated: Van Roy and Wolsey [36] presented a class of flow cover inequalities if $|K_r| = 1$ for $r \in R$ in (ND*). Assuming that $C^+ \neq \emptyset$ and $\bar{u} \coloneqq \max\{u_j : j \in C^+\} > \lambda$, their inequalities are of the form

$$
\begin{aligned}
&\sum_{j\in C^+} [x_j + (u_j - \lambda)^+ \,(1 - y_j)] + \sum_{j\in L^+} [x_j - (\bar{u}_j - \lambda)\, y_j] \\
&\quad \le \beta + \sum_{j\in C^-} [u_j - \min\{\lambda, (u_j - \bar{u} + \lambda)^+\}\,(1 - y_j)] \\
&\quad + \sum_{j\in L^-} \min\{u_j, \max\{u_j - \bar{u} + \lambda, \lambda\}\}\, y_j + \sum_{j\in N^-\setminus(C^-\cup L^-)} x_j,
\end{aligned}
\tag{8.4}
$$

where $\bar{u}_j \coloneqq \max\{\bar{u}, u_j\}$, $L^+ \subseteq N^+ \setminus C^+$, and $L^- \subseteq N^- \setminus C^-$. We will use this later. Moreover, Wolsey [45] derived a class of inequalities for the more general case that $|K_r| \ge 1$, but $K_r \subseteq N^+$ or $K_r \subseteq N^-$ for all $r \in R$. However, these inequalities may be weak w.r.t. the projection to the x-variables: if $N^- = \emptyset$, then they reduce to the form

$$\sum_{r\in U^+} \sum_{j\in K_r} \min\left\{1, \frac{u_{\ell(r)}}{u_j}\right\} x_j + \sum_{r\in U^+} (u_{\ell(r)} - \lambda)^+ \,(1 - \sum_{j\in K_r} y_j) \le \beta, \tag{8.5}$$

where $U^+ \coloneqq \{r \in R : |K_r \cap C^+| = 1\}$ and $\ell(r)$ is the unique element in $K_r \cap C^+$. By Corollary 7.2, the (unique) projection of (8.5) to the x-variables is $\sum_{r\in U^+} \sum_{j\in K_r} \min\{1, \frac{u_{\ell(r)}}{u_j}\}\, x_j \le \beta$, which is redundant for the LP-relaxation of (ND).

In recent work, de Farias et al. [16] present a class of flow cover cuts considering the case that $N^- = \emptyset$. These inequalities are facet defining for P_{ND} in certain cases. In the following, we generalize the work of de Farias et al. [16] to the case in which $N^- \neq \emptyset$ is allowed.

To state our result, we need the following definitions (U^+ is defined as above):

$$
\begin{aligned}
U^+ &\coloneqq \{r \in R : |K_r \cap C^+| = 1\}, &\qquad W^+ &\coloneqq N^+ \setminus \bigcup_{r\in U^+} K_r, \\
U^- &\coloneqq \{r \in R : |K_r \cap C^-| = 1\}, &\qquad W^- &\coloneqq N^- \setminus \bigcup_{r\in U^-} K_r.
\end{aligned}
\tag{8.6}
$$

Note that by assumption, $U^+ \cap U^- = \emptyset$ and $W^+ \cap W^- = \emptyset$. Moreover, for each $r \in U^+ \cup U^-$, we define the sets

$$K_r^{(1)} \coloneqq \{j \in K_r : u_j \ge u_{\ell(r)}\}, \qquad K_r^{(2)} \coloneqq K_r \setminus K_r^{(1)},$$

where $\ell(r)$ is the unique element in $K_r \cap (C^+ \cup C^-)$.

In the following proposition, we restrict ourselves to the case in which either $K_r \subseteq N^+$ or $K_r \subseteq N^-$ for $r \in R$, which is the assumption also made by Wolsey [45]. Later, we will investigate the more general case that some of the K_r can have a nonempty intersection with more than one of the sets N^+, N^-, and $V \setminus (N^+ \cup N^-)$. Moreover, we will show that these inequalities define facets, under certain conditions.

Proposition 8.2. *Let (C^+, C^-) with $C^+ \subseteq N^+$ and $C^- \subseteq N^-$ be some flow cover with $C^+ \neq \emptyset$ and $\bar{u} := \max\{u_j : j \in C^+\} > \lambda$. Furthermore, let $V = \dot{\bigcup}_{r\in R} K_r$ and either $K_r \subseteq N^+$ or $K_r \subseteq N^-$ for $r \in R$. Then*

$$\begin{aligned}
&\sum_{r\in U^+} \sum_{j\in K_r^{(1)}} \frac{u_{\ell(r)}}{u_j} x_j + \sum_{r\in U^+} \sum_{j\in K_r^{(2)}} \max\left\{1, \frac{u_{\ell(r)} - \lambda}{u_j}\right\} x_j \\
&\quad \leq \beta + \sum_{r\in U^-} u_{\ell(r)} + \sum_{r\in U^-} \sum_{j\in K_r^{(1)}} \left(1 - \frac{u_{\ell(r)}}{u_j}\right) x_j \qquad (8.7)\\
&\quad + \sum_{r\in U^-} \sum_{j\in K_r^{(2)}} \left(1 - \max\left\{1, \min\left\{\frac{\lambda}{u_j}, \frac{u_{\ell(r)} - \bar{u} + \lambda}{u_j}\right\}\right\}\right) x_j + \sum_{j\in W^-} x_j
\end{aligned}$$

is valid for P_{ND}.

Proof. We divide the proof into two steps: The first step consists in deriving a valid inequality for (ND*). For this, we adapt a proof of Wolsey [45] with modifications at several places. The second step consists in projecting the inequality for (ND*) to a valid inequality for (ND) with the help of Corollary 7.2.

Since the cliques K_r, $r \in R$ partition V, $K_r \subseteq N^+$ for all $r \in U^+$, and $K_r \subseteq N^-$ for all $r \in U^-$, we obtain that

$$\begin{aligned}
\sum_{i\in N^+} x_i &= \sum_{r\in U^+} \sum_{j\in K_r^{(1)}} x_j + \sum_{r\in U^+} \sum_{j\in K_r^{(2)}} x_j + \sum_{j\in W^+} x_j, \\
\sum_{i\in N^-} x_i &= \sum_{r\in U^-} \sum_{j\in K_r^{(1)}} x_j + \sum_{r\in U^-} \sum_{j\in K_r^{(2)}} x_j + \sum_{j\in W^-} x_j.
\end{aligned}$$

Using the notation $\xi_r := \sum_{j\in K_r^{(1)}} \frac{u_{\ell(r)}}{u_j} x_j$ and $\phi_r := \sum_{j\in K_r^{(1)}} (1 - \frac{u_{\ell(r)}}{u_j}) x_j$ for $r \in U^+ \cup U^-$, we may rewrite

$$\sum_{r\in U^+} \sum_{j\in K_r^{(1)}} x_j = \sum_{r\in U^+} \xi_r + \sum_{r\in U^+} \phi_r, \qquad \sum_{r\in U^-} \sum_{j\in K_r^{(1)}} x_j = \sum_{r\in U^-} \xi_r + \sum_{r\in U^-} \phi_r.$$

Then $\sum_{i\in N^+} x_i - \sum_{i\in N^-} x_i \leq \beta$ is equivalent to

$$\begin{aligned}
&\sum_{r\in U^+} \xi_r + \sum_{r\in U^+} \phi_r + \sum_{r\in U^+} \sum_{j\in K_r^{(2)}} x_j + \sum_{j\in W^+} x_j \leq \beta \\
&\quad + \sum_{r\in U^-} \xi_r + \sum_{r\in U^-} \phi_r + \sum_{r\in U^-} \sum_{j\in K_r^{(2)}} x_j + \sum_{j\in W^-} x_j. \qquad (8.8)
\end{aligned}$$

Furthermore, we introduce auxiliary binary variables $z_r := \sum_{j\in K_r^{(1)}} y_j$ for all $r \in U^+ \cup U^-$. As a relaxation of (ND*), we replace $\sum_{j\in K_r} y_j \leq 1$ by $\xi_r \leq u_{\ell(r)} z_r$ and $\phi_r \leq (u_j - u_{\ell(r)}) z_r$ for all $r \in U^+ \cup U^-$ and remove the same condition $\sum_{j\in K_r} y_j \leq 1$ for all $r \in R \setminus (U^+ \cup U^-)$. This yields:

$$\begin{aligned}
&z_r = \sum_{j \in K_r^{(1)}} y_j && \forall\, r \in U^+ \cup U^-,\\
&0 \le \xi_r \le u_{\ell(r)}\, z_r && \forall\, r \in U^+ \cup U^-,\\
&0 \le \phi_r \le (u_j - u_{\ell(r)})\, z_r && \forall\, r \in U^+ \cup U^-,\\
&0 \le x_j \le u_j\, y_j && \forall\, j \in V,\\
&y \in \{0,1\}^V,\ z \in \{0,1\}^{U^+ \cup U^-}.
\end{aligned} \tag{8.9}$$

We apply Inequality (8.4) to (8.8)–(8.9), where we use the indices corresponding to the variables ξ_r, $r \in U^+ \cup U^-$, as a flow cover; this is possible, since

$$\sum_{r \in U^+} u_{\ell(r)} - \sum_{r \in U^-} u_{\ell(r)} - \beta = \sum_{i \in C^+} u_i - \sum_{i \in C^-} u_i - \beta = \lambda > 0.$$

For simplicity, we further set $L^+ = L^- = \emptyset$ in (8.4); inequalities with $L^+ \neq \emptyset$ and $L^- \neq \emptyset$ can be derived analogously. As a result, we get that

$$\begin{aligned}
\sum_{r \in U^+} \xi_r + \sum_{r \in U^+} (u_{\ell(r)} - \lambda)^+ (1 - z_r) \le \beta + \sum_{r \in U^-} u_{\ell(r)}\\
- \sum_{r \in U^-} \min\left\{\lambda, (u_{\ell(r)} - \bar{u} + \lambda)^+\right\} (1 - z_r) + \sum_{r \in U^-} \phi_r + \sum_{r \in U^-} \sum_{j \in K_r^{(2)}} x_j + \sum_{j \in W^-} x_j
\end{aligned}$$

is valid for (ND*).

To obtain a valid inequality for (ND), we perform a projection to the x-variables space: Recall that $z_r + \sum_{j \in K_r^{(2)}} y_j = \sum_{j \in K_r} y_j \le 1$ for all $r \in U^+ \cup U^-$. The projection works analogously to the proof of Corollary 7.2 by underestimating $1 - z_r \ge \sum_{j \in K_r^{(2)}} y_j \ge \sum_{j \in K_r^{(2)}} \frac{x_j}{u_j}$ for $r \in U^+ \cup U^-$:

$$\begin{aligned}
\sum_{r \in U^+} \xi_r + \sum_{r \in U^+} \sum_{j \in K_r^{(2)}} \frac{(u_{\ell(r)} - \lambda)^+}{u_j} x_j \le \beta + \sum_{r \in U^-} u_{\ell(r)}\\
+ \sum_{r \in U^-} \sum_{j \in K_r^{(2)}} \left(1 - \min\left\{\frac{\lambda}{u_j}, \frac{(u_{\ell(r)} - \bar{u} + \lambda)^+}{u_j}\right\}\right) x_j + \sum_{r \in U^-} \phi_r + \sum_{j \in W^-} x_j.
\end{aligned} \tag{8.10}$$

This inequality may be further strengthened as follows: Given arbitrary sets $Z_r \subseteq K_r^{(2)}$ for $r \in U^+ \cup U^-$, the approach described above works as well using $\overline{K}_r^{(2)} = K_r^{(2)} \setminus Z_r$ instead of $K_r^{(2)}$ and $\bar{\xi}_r := \xi_r + \sum_{j \in Z_r} x_j$ instead of ξ_r. This is due to the fact that the condition $\bar{\xi}_r \le u_{\ell(r)}\, z_r$ is still satisfied. Note, however, that $K_r^{(1)}$ remains unchanged. To obtain lifted coefficients of the variables x_j, $j \in K_r^{(2)}$, in (8.10), we choose Z_r as follows: For $r \in U^+$, we set $Z_r = \{j \in K_r^{(2)} : u_j \ge u_{\ell(r)} - \lambda\}$ such that

$$\begin{aligned}
\sum_{r \in U^+} \bar{\xi}_r + \sum_{r \in U^+} \sum_{j \in \overline{K}_r^{(2)}} \frac{(u_{\ell(r)} - \lambda)^+}{u_j} x_j &= \sum_{r \in U^+} \xi_r + \sum_{r \in U^+} \sum_{j \in Z_r} x_j + \sum_{r \in U^+} \sum_{j \in \overline{K}_r^{(2)}} \frac{(u_{\ell(r)} - \lambda)^+}{u_j} x_j\\
&= \sum_{r \in U^+} \xi_r + \sum_{r \in U^+} \sum_{j \in K_r^{(2)}} \max\left\{1, \frac{u_{\ell(r)} - \lambda}{u_j}\right\} x_j.
\end{aligned}$$

Likewise, we observe that $Z_r = \left\{j \in K_r^{(2)} : u_j \ge \min\left\{\lambda, (u_{\ell(r)} - \bar{u} + \lambda)^+\right\}\right\}$ is the best choice for $r \in U^-$:

$$\begin{aligned}
&\sum_{r\in U^-} 0\cdot\bar{\xi}_r + \sum_{r\in U^-}\sum_{j\in \overline{K}_r^{(2)}} \left(1-\min\left\{\frac{\lambda}{u_j}, \frac{(u_{\ell(r)}-\bar{u}+\lambda)^+}{u_j}\right\}\right) x_j \\
&= \sum_{r\in U^-} 0\cdot\xi_r + \sum_{r\in U^-}\sum_{j\in Z_r} 0\cdot x_j + \sum_{r\in U^-}\sum_{j\in \overline{K}_r^{(2)}} \left(1-\min\left\{\frac{\lambda}{u_j}, \frac{(u_{\ell(r)}-\bar{u}+\lambda)^+}{u_j}\right\}\right) x_j \\
&= \sum_{r\in U^-} 0\cdot\xi_r + \sum_{r\in U^-}\sum_{j\in K_r^{(2)}} \left(1-\max\left\{1,\min\left\{\frac{\lambda}{u_j}, \frac{u_{\ell(r)}-\bar{u}+\lambda}{u_j}\right\}\right\}\right) x_j .
\end{aligned}$$

We obtain the following lifted version of (8.10):

$$\begin{aligned}
&\sum_{r\in U^+} \xi_r + \sum_{r\in U^+}\sum_{j\in K_r^{(2)}} \max\left\{1, \frac{u_{\ell(r)}-\lambda}{u_j}\right\} x_j \le \beta + \sum_{r\in U^-} u_{\ell(r)} \\
&\quad + \sum_{r\in U^-}\sum_{j\in K_r^{(2)}} \left(1-\max\left\{1,\min\left\{\frac{\lambda}{u_j}, \frac{u_{\ell(r)}-\bar{u}+\lambda}{u_j}\right\}\right\}\right) x_j + \sum_{r\in U^-} \phi_r + \sum_{j\in W^-} x_j .
\end{aligned}$$

Resubstitution of $\xi_r = \sum_{j\in K_r^{(1)}} \frac{u_{\ell(r)}}{u_j} x_j$ and $\phi_r = \sum_{j\in K_r^{(1)}} (1-\frac{u_{\ell(r)}}{u_j})\, x_j$ for all $r \in U^+$ yields (8.7). □

We now generalize the flow cover inequalities of Proposition 8.2 to the case in which some of the K_r, $r \in R$, have a nonempty intersection with more than one of the sets N^+, N^-, and $V \setminus (N^+ \cup N^-)$. In fact, we just have to consider one additional term in the formula for each of the two sets $\bigcup_{r\in U^+} K_r \setminus N^+$ and $\bigcup_{r\in U^-} K_r \setminus N^-$.

Theorem 8.3. *Let (C^+, C^-) with $C^+ \subseteq N^+$ and $C^- \subseteq N^-$ be some flow cover with $C^+ \neq \emptyset$ and $\bar{u} := \max\{u_j : j \in C^+\} > \lambda$. Furthermore, let $V = \dot{\bigcup}_{r\in R} K_r$. Then*

$$\begin{aligned}
&\sum_{r\in U^+}\sum_{j\in K_r^{(1)}\cap N^+} \frac{u_{\ell(r)}}{u_j} x_j + \sum_{r\in U^+}\sum_{j\in K_r^{(2)}\cap N^+} \max\left\{1, \frac{u_{\ell(r)}-\lambda}{u_j}\right\} x_j + \sum_{r\in U^+}\sum_{j\in K_r\setminus N^+} \frac{(u_{\ell(r)}-\lambda)^+}{u_j} x_j \\
&\le \beta + \sum_{r\in U^-} u_{\ell(r)} + \sum_{r\in U^-}\sum_{j\in K_r^{(2)}\cap N^-} \left(1-\max\left\{1,\min\left\{\frac{\lambda}{u_j}, \frac{u_{\ell(r)}-\bar{u}+\lambda}{u_j}\right\}\right\}\right) x_j \\
&\quad + \sum_{r\in U^-}\sum_{j\in K_r^{(1)}\cap N^-} \left(1-\frac{u_{\ell(r)}}{u_j}\right) x_j - \sum_{r\in U^-}\sum_{j\in K_r\setminus N^-} \min\left\{\frac{\lambda}{u_j}, \frac{(u_{\ell(r)}-\bar{u}+\lambda)^+}{u_j}\right\} x_j + \sum_{j\in W^-} x_j
\end{aligned}$$

is valid for P_{ND}.

The proof is analogous to the proof of Proposition 8.2.

Remark 8.4. A proof idea in Stallaert [42] gives rise to a further class of valid inequalities for P_{ND}, the so-called *flow pack inequalities*. A *flow packing* is a pair (C^+, C^-) consisting of subsets $C^+ \subseteq N^+$ and $C^- \subseteq N^-$ with $\mu := -\sum_{i\in C^+} u_i + \sum_{i\in C^-} u_i + \beta > 0$. To derive a valid flow pack inequality, we introduce a slack variable $s \ge 0$ to the constraint $\sum_{i\in N^+} x_i + \sum_{i\in N^-} x_i \le \beta$. The resulting equality constraint is multiplied with -1 and relaxed to $\sum_{i\in N^-} x_i - \sum_{i\in N^+} x_i - s \le -\beta$. Since every flow cover of the relaxed constraint is a flow packing of the original one, we may apply Theorem 8.3 to this relaxation to obtain a class of flow pack inequalities corresponding to the presented class of flow cover inequalities.

We now specify sufficient conditions under which the flow cover inequalities of Proposition 8.2 define facets. To state and prove our result, we define

$$Y^+ := \bigcup_{r \in U^+} K_r^{(2)} \qquad\qquad Y^- := \bigcup_{r \in U^-} K_r^{(2)}. \tag{8.11}$$

Theorem 8.5. *Let $V = N^+ \cup N^-$ and let either $K_r \subseteq N^+$ or $K_r \subseteq N^-$ for all $r \in R$. Furthermore, assume that $V = \dot\bigcup_{r\in R} K_r$, and $\beta > 0$ or $N^- \neq \emptyset$. Then Inequality (8.7) defines a facet of P_{ND} if*

(i) $|K_r^{(1)}| = 1$ *for all* $r \in U^+ \cup U^-$,

(ii) $x_{C^-} = u_{C^-}$ *and* $x_{Y^-} = 0$ *are fixed, and*

(iii) $u_\nu < u_{\ell(r^*)} - \lambda$ *for some* $r^* \in U^+$ *and* $\nu \in K_{r^*}^{(2)}$.

Proof. The proof is based on standard methods like in Van Roy and Wolsey [36].

By assumption, we have that $N^+ \supseteq \bigcup_{r\in U^+} K_r$ and $N^- \supseteq \bigcup_{r\in U^-} K_r$. Further, notice that from *(i)* and the definition of U^+, U^- in (8.6), we obtain $C^+ = \bigcup_{r\in U^+} K_r^{(1)}$ and $C^- = \bigcup_{r\in U^-} K_r^{(1)}$. The sets N^+ and N^- are partitioned into $C^+ \,\dot\cup\, Y^+ \,\dot\cup\, W^+$ and $C^- \,\dot\cup\, Y^- \,\dot\cup\, W^-$, respectively, where Y^+, Y^- are defined as in (8.11) and W^+, W^- as in (8.6).

Let

$$M_1 := \{i \in C^+ \,:\, u_i \geq \lambda\}, \qquad M_3 := \{i \in Y^+ \,:\, u_i < u_{\ell(r_i)} - \lambda\},$$
$$M_2 := \{i \in C^+ \,:\, u_i < \lambda\}, \qquad M_4 := \{i \in Y^+ \,:\, u_i \geq u_{\ell(r_i)} - \lambda\},$$

where r_i, $i \in Y^+$, is the unique index $r \in U^+$ with $i \in K_r$. In order to show the facet-defining property, we constitute dimension of P_{ND} many affinely independent points that are feasible for (ND) and satisfy Inequality (8.7) with equality. The dimension of P_{ND} is $|N^+ \cup N^-| - |C^-| - |Y^-|$, since the subvectors x_{C^-} and x_{Y^-} are fixed by *(ii)*, and since $u > 0$ as well as $\beta > 0$ or $N^- \neq \emptyset$. The latter conditions guarantee that the unit vectors are feasible under an appropriate scaling. For $\varepsilon > 0$ sufficiently small, the points read as follows:

$$\begin{array}{rcl}
 & & \overbrace{\hphantom{u_{C^+} - (\lambda-\varepsilon)\, e_{\ell(r^*)} - \varepsilon\, e_i}}^{C^+} \quad \overbrace{\hphantom{(u_{\ell(r_i)} - \lambda)\, e_i}}^{Y^+} \quad \overbrace{\hphantom{\varepsilon e_i}}^{W^+} \quad \overbrace{\hphantom{u_{C^-}}}^{C^-} \quad \overbrace{\hphantom{0}}^{Y^-} \quad \overbrace{\hphantom{\varepsilon e_i}}^{W^-} \\
p^i &=& (\; u_{C^+} - \lambda\, e_i \;,\; 0 \;,\; 0 \;,\; u_{C^-} \;,\; 0 \;,\; 0 \;) \quad \forall\, i \in M_1, \\
\bar p^i &=& (\; u_{C^+} - (\lambda - \varepsilon)\, e_{\ell(r^*)} - \varepsilon\, e_i \;,\; 0 \;,\; 0 \;,\; u_{C^-} \;,\; 0 \;,\; 0 \;) \quad \forall\, i \in M_2, \\
q^i &=& (\; u_{C^+} - u_{\ell(r_i)}\, e_{\ell(r_i)} \;,\; u_i\, e_i \;,\; 0 \;,\; u_{C^-} \;,\; 0 \;,\; 0 \;) \quad \forall\, i \in M_3, \\
\bar q^i &=& (\; u_{C^+} - u_{\ell(r_i)}\, e_{\ell(r_i)} \;,\; (u_{\ell(r_i)} - \lambda)\, e_i \;,\; 0 \;,\; u_{C^-} \;,\; 0 \;,\; 0 \;) \quad \forall\, i \in M_4, \\
s^i &=& (\; u_{C^+} - u_{\ell(r^*)}\, e_{\ell(r^*)} \;,\; u_\nu\, e_\nu \;,\; \varepsilon\, e_i \;,\; u_{C^-} \;,\; 0 \;,\; 0 \;) \quad \forall\, i \in W^+, \\
\bar s^i &=& (\; u_{C^+} - (\lambda - \varepsilon)\, e_{\ell(r^*)} \;,\; 0 \;,\; 0 \;,\; u_{C^-} \;,\; 0 \;,\; \varepsilon\, e_i \;) \quad \forall\, i \in W^-,
\end{array} \tag{8.12}$$

where $r^* \in U^+$, $\nu \in K_{r^*}^{(2)}$ are indices satisfying the condition in *(iii)*. Since $C^+ = \bigcup_{r\in U^+} K_r^{(1)}$, $C^- = \bigcup_{r\in U^-} K_r^{(1)}$, and the sets $K_r^{(1)}$ for $r \in U^+ \cup U^-$ are singletons (see *(i)*), we have that

$$\sum_{r\in U^+} \sum_{j \in K_r^{(1)}} \frac{u_{\ell(r)}}{u_j} x_j = \sum_{i\in C^+} x_i \qquad \text{and} \qquad \sum_{r \in U^-} u_{\ell(r)} = \sum_{i \in C^-} u_i. \tag{8.13}$$

Inserting the points (8.12) shows that they satisfy Inequality (8.7) with equality. In the following, we prove that the points are feasible for (ND) and affinely independent.

The feasibility of p^i, q^i, and $\bar q^i$ for (ND) follows directly from the definition $\lambda := \sum_{i\in C^+} u_i - \sum_{i \in C^-} u_i - \beta$ and the restrictions of the sets M_1, M_3, and M_4, respectively. The feasibility of s^i follows with *(iii)*. Since by *(iii)*, we also derive $u_{\ell(r^*)} > \lambda$, we get the feasibility of $\bar p^i$ and $\bar s^i$.

It remains to show that the points (8.12) are affinely independent: We build a linear equation system, where each equation is obtained by inserting one of the points in (8.12) for x into

$$\sum_{j\in C^+} \pi_j\, x_j + \sum_{j\in Y^+} \pi_j\, x_j + \sum_{j\in W^+} \pi_j\, x_j = \pi_0 + \sum_{j\in C^-} \pi_j\, x_j + \sum_{j\in Y^-} \pi_j\, x_j + \sum_{j\in W^-} \pi_j\, x_j. \tag{8.14}$$

The points are affinely independent if all solutions (π, π_0) to this linear equality system are uniquely represented by the coefficient vector of the flow cover inequality (8.7) up to a scalar multiple.

We first project the variables x_{C^-} and x_{Y^-} out of the problem. Since these variables are fixed by *(ii)*, this only leads to a new constant $\bar{\pi}_0$:

$$\sum_{j\in C^+} \pi_j\, x_j + \sum_{j\in Y^+} \pi_j\, x_j + \sum_{j\in W^+} \pi_j\, x_j = \bar{\pi}_0 + \sum_{j\in W^-} \pi_j\, x_j. \tag{8.15}$$

Further, from the points p^i and $\bar{p}^i$, we get

$$\begin{aligned} &\sum_{j\in C^+} \pi_j\, u_j - \pi_i\, \lambda = \bar{\pi}_0 && \forall\, i \in M_1,\\ &\sum_{j\in C^+} \pi_j\, u_j - \pi_{\ell(r^*)}\,(\lambda - \varepsilon) - \pi_i\, \varepsilon = \bar{\pi}_0 && \forall\, i \in M_2. \end{aligned}$$

The validity of the first equation shows that the coefficients π_i, $i \in M_1$, are all equal to $(\sum_{j\in C^+} \pi_j\, u_j - \bar{\pi}_0)/\lambda =: \bar{\pi}$. Since in particular, $\pi_{\ell(r^*)} = \bar{\pi}$ (by *(iii)*, $\ell(r^*) \in M_1$), we derive from the second equation that $\pi_i = \bar{\pi}$ for all $i \in M_2$. Moreover, because $C^+ = M_1 \dot{\cup} M_2$, we obtain that $\bar{\pi}_0 = \bar{\pi}\,(\sum_{j\in C^+} u_j - \lambda)$. Inserting q^i and $\bar{q}^i$ in (8.15) gives

$$\begin{aligned} &\sum_{j\in C^+\setminus\{\ell(r_i)\}} \pi_j\, u_j + \pi_i\, u_i = \bar{\pi}_0 && \forall\, i \in M_3,\\ &\sum_{j\in C^+\setminus\{\ell(r_i)\}} \pi_j\, u_j + \pi_i\,(u_{\ell(r_i)} - \lambda) = \bar{\pi}_0 && \forall\, i \in M_4. \end{aligned}$$

Using that $\bar{\pi}_0 = \bar{\pi}\,(\sum_{j\in C^+} u_j - \lambda)$ and $\pi_j = \bar{\pi}$ for $j \in C^+$, we get $\pi_i = \bar{\pi}\,\frac{u_{\ell(r_i)}-\lambda}{u_i}$ for $i \in M_3$ from the first equation and $\pi_i = \bar{\pi}$ for $i \in M_4$ from the second equation. Finally, from s^i and $\bar{s}^i$, we derive

$$\begin{aligned} &\sum_{j\in C^+\setminus\{\ell(r^*)\}} \pi_j\, u_j + \pi_\nu\, u_\nu + \pi_i\, \varepsilon = \bar{\pi}_0 && \forall\, i \in W^+,\\ &\sum_{j\in C^+} \pi_j\,(u_j - \lambda + \varepsilon) = \bar{\pi}_0 + \pi_i\, \varepsilon && \forall\, i \in W^-. \end{aligned}$$

By (iii), we have that $\nu \in M_3$, and hence $\pi_\nu = \bar{\pi}\,\frac{u_{\ell(r^*)}-\lambda}{u_\nu}$, as seen above. We substitute $\bar{\pi}_0 = \bar{\pi}\,(\sum_{j\in C^+} u_j - \lambda)$, $\pi_\nu = \bar{\pi}\,\frac{u_{\ell(r^*)}-\lambda}{u_\nu}$, and $\pi_j = \bar{\pi}$ for $j \in C^+$ into the first equation, to obtain $\pi_i = 0$ for $i \in W^+$. Moreover, from the second equation, we obtain $\pi_i = \bar{\pi}$ for $i \in W^-$.

According to the definition $\lambda := \sum_{j\in C^+} u_j - \sum_{j\in C^-} u_j - \beta$, it holds that $\bar{\pi}_0 = \bar{\pi}\,(\sum_{j\in C^+} u_j - \lambda) = \bar{\pi}\,(\beta + \sum_{j\in C^-} u_j)$. Summarizing the results from above, (8.14) is uniquely represented by

$$\sum_{j\in C^+ = M_1 \dot{\cup} M_2} \bar{\pi}\, x_j + \sum_{j\in M_3} \bar{\pi}\, \frac{u_{\ell(r_j)}-\lambda}{u_j}\, x_j + \sum_{j\in M_4} \bar{\pi}\, x_j = \bar{\pi}\left(\beta + \sum_{j\in C^-} u_j\right) + \sum_{j\in W^-} \bar{\pi}\, x_j, \tag{8.16}$$

up to the scalar multiple $\bar{\pi}$. By (8.13) and the definition of M_3 and M_4, it follows that the coefficient vector of (8.16) coincides with the coefficient vector of the flow cover inequality (8.7) up to the multiple $\bar{\pi}$ and for the special case that $x_{Y^-} = 0$, which proves the claim.

$\square$

A separation heuristic for the presented class of flow cover inequalities is presented in [22].

9 Computational Experience

This section reports on computational experience with a branch-and-cut algorithm based on the presented concepts above. Our implementation is written in C using the framework SCIP (see [28] [41]) with CPLEX as LP-solver. The supported branching rules are neighborhood and hyperclique branching. The only cutting planes we implemented are hyperclique bound inequalities (see Section 8.1) and flow cover inequalities (see Section 8.3).

We demonstrate the effectiveness of our implementation on randomly generated instances and instances taken from the literature. The results are compared with the solution of the MIP-solvers CPLEX and SCIP, which are applied both to (MIPPC). Our experiments were run on a Linux cluster with Intel i3 3.2GHz dual core processors and 8GB main memory. We used SCIP 3.2.0 and CPLEX 12.6.1 for the experiments in Section 9.2 and SCIP 3.2.1 and CPLEX 12.6.3 for the experiments in Section 9.3. For our tests, we set a CPU time limit of two hours per instance.

9.1 Instance Sets

For our computational study, we used instances from three different kind of problems that we explain below. The instance collections of the first two problems are used to demonstrate the effect of branching rules (Section 6.1) and hyperclique bound inequalities (Section 8.1). The third collection consists of instances with SOS1 constraints to show the benefit of the flow cover inequalities of Section 8.3.

Cardinality Constrained Multidimensional Knapsack Problem The mathematical formulation of this problem type is

$$\begin{aligned} \text{(CCMKP)} \qquad \max \quad & \sum_{j=1}^{n} c_j\, x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{ij}\, x_j \leq b_i \quad && i \in \{1,\dots,m\}, \\ & 0 \leq x \leq u, \\ & \|x_{K_r}\|_0 \leq k_r && r \in \{1,\dots,q\}, \end{aligned}$$

where n, m, and q are positive integers, $A = (a_{ij}) \in \mathbb{R}_+^{m\times n}$, $b \in \mathbb{R}_+^m$, $u \in (\mathbb{R}_+ \cup \{\infty\})^n$, $c \in \mathbb{R}_+^n$, $K_1,\dots,K_q \subseteq \{1,\dots,n\}$, and $k_1,\dots,k_q$ are integers with $0 < k_r < |K_r|$. This problem was studied by de Farias and Nemhauser [17] for the special case that $m = q = 1$.

We generated CCMKP instance sets of five different types. Table 9.1 reports on the detailed statistics. Each problem type has 25 individual instances (see column "#"). All 125 instances have $n = 10\,000$ variables and $m = 50$ linear constraints. The number of the cardinality constraints is listed in column "q" and can vary among the different types. The objective coefficients c were generated uniformly at random in the interval $[10, 26]$, and the right hand sides b_i were determined according to $b_i = 1/50 \cdot \sum_{j=1}^{n} a_{ij}$. Further, the nonzero entries of the matrix A were chosen at random from $[5, 20] \cap \mathbb{Z}$. The expected matrix density of A is specified in column "dens(A)". The upper bounds u are either all 1 for CCMKP type three and four or all infinity for CCMKP type one, two, and five. Note that in the latter case, finite big-M values for the MIPPC formulation are implied by the linear inequality system. The cardinality constraints of each individual instance all have the same size $|K| = |K_r|$ and right hand side $k = k_r$ for $r \in \{1,\dots,q\}$ (see columns "$|K|$" and "k"). Overlapping hypercliques only appear for CCMKP type five (see column "overlap"); here, the assignment of variables to the hypercliques K_r, $r \in R$, was chosen at random. For the remaining instance types, the variables are assigned to the hypercliques in the order of their indices.

Table 9.1: Statistics of the CCMKP instance sets

Test set	#	n	m	q	dens(A)	u	$\lvert K\rvert$	k	overlap
CCMKP Type 1	25	10000	50	1	0.33	∞	10000	40	no
CCMKP Type 2	25	10000	50	1	0.66	∞	10000	40	no
CCMKP Type 3	25	10000	50	20	0.33	$\mathbb{1}$	500	20	no
CCMKP Type 4	25	10000	50	20	0.33	$\mathbb{1}$	500	18	no
CCMKP Type 5	25	10000	50	100	0.66	∞	800	5	yes

Automated Teller Machine Cash Management Problem This second kind of problem arises from an application in automated teller machine (ATM) cash management, see Galati [26]. We modeled 10 instances with the raw data from [27]. Detailed statistics of each individual instance can be found in Table 9.2. The column labels have the same meaning as in the former table with the following specifications: Column "n (bin)" refers to the total number of variables; the number of binary variables are given in parenthesis. The column labeled by m refers to the number of equality and inequality constraints and the one labeled by k refers to the arithmetic mean over all numbers k_r, $r \in \{1, \dots, q\}$. The data for the upper bounds u of the variables emerge from the application and may differ from variable to variable.

The cardinality constraints of the ATM instances do not overlap. Moreover, since the ATM instances contain binary variables, they are rather MIPs with cardinality constraints and not CCLPs. Nevertheless, all variables that are involved in a cardinality constraint are continuous. We used the existing branching routines of SCIP to enforce the binary conditions, where we put a higher priority on the enforcement of cardinality constraints. This means that the binary conditions are enforced only when all the cardinality constraints are not violated by the current LP solution, which produced the best results for these instances.

Table 9.2: Statistics of the ATM instances (k in arithmetic mean over all cardinality constraints)

Test instance	n (bin)	m	q	$\lvert K\rvert$	k	overlap
`atm_5_25_1`	370 (50)	315	5	25	10.2	no
`atm_5_25_2`	370 (50)	315	5	25	10.6	no
`atm_5_25_3`	370 (50)	315	5	25	9.0	no
`atm_5_25_4`	370 (50)	315	5	25	8.4	no
`atm_5_25_5`	370 (50)	315	5	25	11.2	no
`atm_5_50_1`	620 (50)	465	5	50	16.0	no
`atm_5_50_2`	620 (50)	465	5	50	22.6	no
`atm_5_50_3`	620 (50)	465	5	50	17.0	no
`atm_5_50_4`	620 (50)	465	5	50	20.2	no
`atm_5_50_5`	620 (50)	465	5	50	20.8	no

Maximum Flow Problem with Conflict Constraints Our final instance sets consist of maximum flow problems (MFP) from an application in telecommunications engineering. We used 30 instances taken from [24] in which all variables are integer, and 30 additional instances in which all variables are continuous. We denote the instances from [24] with "MFP of type 1" and the additional instances with "MFP of type two".

The statistics of the MFP instance sets are summarized in Table 9.3. The cardinality constraints of all instances can overlap with each other and they all have right hand side $k = 1$, i.e., they are SOS1 constraints.

Table 9.3: Statistics of the MFP instance sets (in arithmetic mean)

Test set	n (int)	m	q	$\|K\|$	k	overlap
MFP Type 1	2317.4 (2317.4)	79.6	2184.5	27.4	1.0	yes
MFP Type 2	2363.3 (0)	79.6	2166.5	27.9	1.0	yes

Table 9.4: Settings for test runs

shortcut	branching rule	bound cut separation
B-neigh-noC	neighborhood	separation off
B-neigh-C0	neighborhood	only in the root node
B-neigh-C1	neighborhood	with node depth frequency 1
B-neigh-C10	neighborhood	with node depth frequency 10
B-balance-C1	balanced	with node depth frequency 1

9.2 Computational Results for Instances with Cardinality Constrains

Our computational study for cardinality constraints is divided into four different experiments. In Experiment 1, we evaluate the effect of (hyperclique) bound inequalities and in Experiment 2, we compare neighborhood branching with hyperclique branching. For both experiments, we initialized the algorithm with precomputed optimal solutions to eliminate the influence of primal heuristics on the performance. In Experiment 3, we no longer initialize the algorithm with precomputed optimal solutions, but make use of the primal heuristics of SCIP. Finally, in Experiment 4, we compare our implementation with other branch-and-cut solvers.

The parameter settings that we have chosen for our test runs are listed in Table 9.4. Each shortcut specifies which branching rule is used and whether or with which node depth frequency bound inequalities of Section 8.1 are separated. The term “balanced branching” stands for the combination of hyperclique and neighborhood branching, where we apply hyperclique branching only in specific situations as explained in Section 6.1. For balanced branching, we have to separate bound inequalities with node depth frequency one, since otherwise it can be the case that our selection rule in each iteration repeatedly chooses the same constraint for branching, resulting in an infinite loop (see Section 6.1).

The data of Table 9.5 report on the results of all four experiments. We list the number of instances that reached the time limit (column “limit”), the shifted geometric mean of the number of applied bound cuts (column “bnd-cuts”), the number of nodes (column “nodes”), and the CPU time (column “time”). The shifted geometric mean of values $t_1, \dots, t_n$ with shift value δ is defined as $(\Pi_{i=1}^{n}(t_i + \delta))^{1/n} - \delta$. We used a shift of 10 for the number of cuts and the CPU time and 100 for the number of nodes.

Experiment 1: Hyperclique bound inequalities Hyperclique bound inequalities have a high positive impact for CCMKP of type three and four, since for these instances, the upper bounds u are tight, i.e., attainable by a feasible solution (see Table 9.1). Here, the best results were obtained if bound inequalities are separated with node depth frequency 1 (see row “B-neigh-C1”). On the other hand, if the separation of bound inequalities is turned off (see row “B-neigh-noC”), then this significantly increases the CPU time for these instances. For the instances ATM the influence of bound inequalities is negligible and for the instances CCMKP of type one, two, and five, for which the upper bounds u are not tight, no bound cuts were separated, independent of the settings.

Experiment 2: Branching rules We proceed to the comparison of neighborhood and balanced branching in rows "B-neigh-C1" and "B-balance-C1".

For the CCMKP instances of type three and four, balanced branching is much more effective than neighborhood branching. The reason is that for these instances, the upper bounds are tight and the more balanced branching tree results in a reduced number of branching nodes. For balanced branching, the number of separated bound inequalities increase in relation to the number of branching nodes. This is because balanced branching produces additional local cardinality constraints by branching on hypercliques. The local constraints serve for the separation of additional bound inequalities, which can be strong because of the tight upper bounds of the variables.

For the remaining instance types, it does not make a significant difference whether to use neighborhood or balanced branching. The reason is that the selection rule of balanced branching only decides to branch on a hyperclique if the bound inequalities of the produced local cardinality constraints are sufficiently tight, see Section 6.1. Otherwise, neighborhood branching is used.

Experiment 3: Primal heuristics After evaluation of the former experiments we decided to use "B-balance-C1" as our favorite setting. For the experiment in row "B-balance-C1-H", we no longer initialize the algorithm with precomputed optimal solutions, but make use of the primal heuristics of SCIP. Comparing rows "B-balance-C1" and "B-balance-C1-H", it can be seen that our solver requires much more time if optimal solutions are not initialized; especially if the upper bounds of the variables are tight, as it is the case for the CCMKP instances of type three and four. Obviously the heuristics built into SCIP not seem to be effective. For the future, it is therefore advisable to improve the performance with additional heuristics.

Experiment 4: Comparison with other solvers In this final experiment, we compare our implementation (row "B-balance-C1-H") with the MIP-solvers SCIP (row "SCIP-MIP") and CPLEX (row "CPLEX-MIP") in default settings. Our implementation outperforms both MIP-solvers for CCMKP type one, two, and five. However, for CCMKP type three and four and ATM, the solvers SCIP-MIP and CPLEX-MIP are faster. CPLEX is even faster if our implementation is initialized with precomputed optimal solutions, see row "B-balance-C1".

The results indicate that our approach can be beneficial if the upper bounds u are not tight. In this case, bound inequalities are not efficient, and although the LP-relaxation of (CCLP) has less variables and constraints, it is approximately as tight as the LP-relaxation of (MIPPC). On the other hand, if the bounds are tight, then the (MIPPC) formulation should be preferred. It remains an open question how a more balanced branching scheme would perform for (MIPPC).

9.3 Computational Results for Instances with SOS1 Constraints

In this second section of experiments, we demonstrate the benefit of using flow cover inequalities on the SOS1 constrained instance sets MFP of type 1 and 2. We consider two settings: "B-neigh" for turning flow cover inequalities off and "B-neigh-flow" for separating flow cover inequalities in the root node. For both settings, we make use of neighborhood branching and bound inequalities separated in the root node.

The data in Table 9.6 shows the aggregated results of the two MFP instance sets, where column "flow-cuts" denotes the mean number of applied flow cover cuts. The results demonstrate that flow cover inequalities improve the CPU time and allow to solve more instances within the limits. Moreover, the reduction in the CPU time was confirmed by a Wilcoxon signed rank test, see [44], with a p-value of less than 0.025 for MFP of type 1 and 0.005 for MFP of type 2.

Table 9.5: Experiments on the CCMKP Type 1–5 and ATM instances (cuts, nodes, and time in shifted geometric mean)

	CCMKP Type 1 (25)				CCMKP Type 2 (25)			
Setting	limit	bnd-cuts	nodes	time	limit	bnd-cuts	nodes	time
B-neigh-noC	0	0.0	61134.1	1048.1	2	0.0	50137.3	1373.6
B-neigh-C0	0	0.0	61134.1	1048.1	2	0.0	50148.4	1374.0
B-neigh-C10	0	0.0	61134.1	1049.0	2	0.0	50146.2	1373.1
B-neigh-C1	0	0.0	61134.1	1058.8	2	0.0	50122.6	1380.1
B-balance-C1	0	0.0	61134.1	1058.7	2	0.0	50107.4	1382.5
B-balance-C1-H	0	0.0	62021.7	1157.4	2	0.0	51118.3	1489.2
SCIP-MIP	15	–	139270.7	5907.8	12	–	80427.9	5000.9
CPLEX-MIP	2	–	225218.5	1961.2	5	–	205523.8	2668.6

	CCMKP Type 3 (25)				CCMKP Type 4 (25)			
Setting	limit	bnd-cuts	nodes	time	limit	bnd-cuts	nodes	time
B-neigh-noC	3	0.0	52859.5	267.6	20	0.0	1020085.0	5511.1
B-neigh-C0	0	1.2	15653.9	82.8	5	4.3	451610.4	2002.5
B-neigh-C10	0	17.2	15063.1	80.5	3	491.3	393368.1	1764.8
B-neigh-C1	0	19.4	13369.3	74.0	3	751.8	365200.3	1707.8
B-balance-C1	0	114.7	11649.7	67.3	0	683.6	303029.0	1414.8
B-balance-C1-H	0	280.3	53548.5	434.5	21	6884.8	812457.9	6991.3
SCIP-MIP	0	–	1433.8	126.9	0	–	28016.4	991.5
CPLEX-MIP	0	–	832.5	22.6	0	–	20295.1	476.6

	CCMKP Type 5 (25)				ATM (10)			
Setting	limit	bnd-cuts	nodes	time	limit	bnd-cuts	nodes	time
B-neigh-noC	0	0.0	23237.3	494.6	0	0.0	7722.7	24.4
B-neigh-C0	0	0.0	23237.3	495.1	0	0.0	7251.8	24.4
B-neigh-C10	0	0.0	23237.3	496.9	0	0.5	7252.6	24.7
B-neigh-C1	0	0.0	23237.3	511.5	0	1.0	7253.1	24.5
B-balance-C1	0	0.0	23237.3	512.3	0	1.0	7253.1	24.7
B-balance-C1-H	0	0.0	32699.1	837.2	0	1.1	14899.6	40.4
SCIP-MIP	0	–	23423.2	1858.9	0	–	3754.5	27.6
CPLEX-MIP	1	–	55686.1	1514.1	0	–	4036.5	24.5

Table 9.6: Experiments on the MFP instances (cuts, nodes, and time in shifted geometric mean)

	MFP Type 1 (25)				MFP Type 2 (25)			
Setting	limit	flow-cuts	nodes	time	limit	flow-cuts	nodes	time
B-neigh	11	0.0	687501.8	1551.5	11	0.0	1936704.6	2342.1
B-neigh-flow	9	13.0	486232.7	1159.4	7	12.2	1414270.8	1804.9

10 Conclusion

In this article, we investigated LPs with overlapping cardinality constraints and analyzed their complexity status. Moreover, we studied branching rules and cutting planes for a branch-and-cut approach. One result is that (CCLP) can be solved in polynomial time for co-triangulated conflict hypergraphs. In the main part of our investigations, we motivated a new branching rule called dependent set branching, which we derived from a representation of (CCLP) as a disjunctive program. This branching rule can result in a tighter decomposition of the solution set than standard 0/1-branching on the binary variables of a MIP formulation.

In a polyhedral study, we described a procedure to derive valid or facet defining inequalities. We demonstrated this approach by deriving three different classes of cutting planes: hyperclique bound cuts, implied bound cuts, and flow cover cuts. The latter cutting planes generalize the ones presented by de Farias et al. [16] to the case that the constraint matrix has both positive and negative entries.

Experimental results show the behavior of our implementation on instances of three different applications. We compared the performance of neighborhood branching, which can result in an unbalanced branch-and-bound tree, to a more balanced branching rule, which makes use of hyperclique branching. To our knowledge, hyperclique branching has not been tested in practice before and only theoretically investigated by Farias et al. [14]. We substantiated that balanced branching can be beneficial if the bounds of the variables are tight. Otherwise, pure neighborhood branching is a better choice for the instances we considered. The cutting planes we implemented are hyperclique bound cuts and flow cover cuts. In case that a significant number of them was applied, they both always could reduce the solution time. We also compared the results of our implementation with the solution of the solvers CPLEX and SCIP, which we both applied to a MIP reformulation. The results indicate that if the bounds of the variables are not tight, then cardinality constraints should be handled directly without introducing auxiliary binary variables to the model. On the other hand, if the bounds are tight, then a MIP reformulation should be preferred.

Acknowledgments

This work was supported by the EXPRESS project within the DFG priority program on Compressed Sensing in Information Processing (CoSIP DFG-SPP 1798) and the Excellence Initiative of the Graduate School of Computational Engineering at Technische Universität Darmstadt.

References

[1] T. Achterberg. SCIP: Solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41, 2009.

[2] J. A. Appleget and R. K. Wood. Explicit-constraint branching for solving mixed-integer programs. In *Computing Tools for Modeling, Optimization and Simulation*, pages 245–261. Springer Verlag, 2000.

[3] E. Balas. Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics*, 89(1–3):3–44, 1998.

[4] E. Balas and S. M. Ng. On the set covering polytope: I. all the facets with coefficients in $\{0, 1, 2\}$. *Mathematical Programming*, 43(1):57–69, 1989.

[5] E. Balas and M. Oosten. On the dimension of projected polyhedra. *Discrete Applied Mathematics*, 87(1–3):1–9, 1998.

[6] E. M. L. Beale and J. A. Tomlin. Special facilities in general mathematical programming system for non-convex problems using ordered sets of variables. In J. Lawrence, editor, *Proc. 5th International Conference on Operations Research*, pages 447–454. Travistock Publications, London, 1970.

[7] C. Berge. *Hypergraphs: Combinatorics of Finite Sets.* North-Holland, Amsterdam, 1984.

[8] D. Bienstock. Computational study of a family of mixed-integer quadratic programming problems. In E. Balas and J. Clausen, editors, *Integer Programming and Combinatorial Optimization*, volume 920 of *Lecture Notes in Computer Science*, pages 80–94. Springer Verlag Berlin Heidelberg, 1995.

[9] E. Boros, K. Elbassioni, V. Gurvich, and L. Khachiyan. Generating maximal independent sets for hypergraphs with bounded edge-intersections. In M. Farach-Colton, editor, *LATIN 2004: Theoretical Informatics*, volume 2976 of *Lecture Notes in Computer Science*, pages 488–498. Springer Verlag Berlin Heidelberg, 2004.

[10] O. P. Burdakov, C. Kanzow, and A. Schwartz. Mathematical programs with cardinality constraints: Reformulation by complementarity-type conditions and a regularization method. *SIAM Journal on Optimization*, 26(1):397–425, 2016.

[11] S. J. Chung. NP-completeness of the linear complementarity problem. *Journal of Optimization Theory and Applications*, 60:393–399, 1989.

[12] M. Conforti, G. Cornuéjols, and G. Zambelli. Polyhedral approaches to mixed integer linear programming. In M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, editors, *50 Years of Integer Programming*, pages 343–385. Springer Verlag Berlin Heidelberg, 2010.

[13] E. Dahlhaus. Minimal elimination ordering inside a given chordal graph. In R. H. Möhring, editor, *Graph-Theoretic Concepts in Computer Science*, volume 1335 of *Lecture Notes in Computer Science*, pages 132–143. Springer Verlag Berlin Heidelberg, 1997.

[14] I. R. de Farias, E. L. Johnson, and G. L. Nemhauser. Branch-and-cut for combinatorial optimization problems without auxiliary binary variables. *Knowl. Eng. Rev.*, 16(1):25–39, 2001.

[15] I. R. de Farias and E. Kozyreff. A note on critical-set and lifted surrogate inequalities for cardinality-constrained linear programs. *Computers and Industrial Engineering*, 82:1–7, 2015.

[16] I. R. de Farias, E. Kozyreff, and M. Zhao. Branch-and-cut for complementarity-constrained optimization. *Mathematical Programming Computation*, pages 1–39, 2014.

[17] I. R. de Farias and G. L. Nemhauser. A polyhedral study of the cardinality constrained knapsack problem. *Mathematical Programming*, 96(3):439–467, 2003.

[18] P. M. Dearing, D. R. Shier, and D. D. Warner. Maximal chordal subgraphs. *Discrete Applied Mathematics*, 20(3):181–190, 1988.

[19] T. Easton, K. Hooker, and E. K. Lee. Facets of the independent set polytope. *Mathematical Programming*, 98(1–3):177–199, 2003.

[20] E. Emtander. A class of hypergraphs that generalizes chordal graphs. *Mathematica Scandinavica*, 106(1):50–66, 2010.

[21] R. Euler, M. Jünger, and G. Reinelt. Generalizations of cliques, odd cycles and anticycles and their relation to independence system polyhedra. *Mathematics of Operations Research*, 12(3):451–462, 1987.

[22] T. Fischer. *Branch-and-cut for complementarity and cardinality constrained linear programs*. PhD thesis, Technical University of Darmstadt, 2017.

[23] T. Fischer and M. E. Pfetsch. Branch-and-cut for linear programs with overlapping SOS1 constraints. Technical report, Available on Optimization Online, submitted for publication, 2015.

[24] T. Fischer and M. E. Pfetsch. Monoidal cut strengthening and generalized mixed-integer rounding for disjunctive programs. Technical report, 2016.

[25] D. Gade and S. Küçükyavuz. Formulations for dynamic lot sizing with service levels. *Naval Research Logistics*, 60(2):87–101, 2013.

[26] M. Galati. *Decomposition methods for integer linear programming*. PhD thesis, Lehigh University, 2010.

[27] M. Galati and T. Ralphs. `http://www.coin-or.org/projects/Dip.xml`, DIP, Decomposition for integer programming.

[28] G. Gamrath, T. Fischer, T. Gally, A. M. Gleixner, G. Hendel, T. Koch, S. J. Maher, M. Miltenberger, B. Müller, M. E. Pfetsch, C. Puchert, D. Rehfeldt, S. Schenker, R. Schwarz, F. Serrano, Y. Shinano, S. Vigerske, D. Weninger, M. Winkler, J. T. Witt, and J. Witzig. The SCIP optimization suite 3.2. ZIB Report 15-60, 2016.

[29] J. Gao and D. Li. Optimal cardinality constrained portfolio selection. *Operations Research*, 61(3):745–761, 2013.

[30] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*. Algorithms and combinatorics. Springer Verlag, 1988.

[31] T. Hoheisel, C. Kanzow, and A. Schwartz. Theoretical and numerical comparison of relaxation methods for mathematical programs with complementarity constraints. *Mathematical Programming*, 137:257–288, 2013.

[32] R. G. Jeroslow. Representability in mixed integer programming, I: Characterization results. *Discrete Applied Mathematics*, 17(3):223–243, 1987.

[33] M. Laurent. A generalization of antiwebs to independence systems and their canonical facets. *Mathematical Programming*, 45(1–3):97–108, 1989.

[34] S. Maheshwary. *Facets of conflict hypergraphs*. PhD thesis, Georgia Institute of Technology, 2008.

[35] K. G. Murty. *Linear complementarity, linear and nonlinear programming*. Sigma series in applied mathematics. Heldermann Verlag, Berlin, 1988.

[36] T. J. V. Roy and L. A. Wolsey. Valid inequalities for mixed 0-1 programs. *Discrete Applied Mathematics*, 14(2):199–213, 1986.

[37] D. M. Ryan and B. A. Foster. An integer programming approach to scheduling. In A. Wren, editor, *Computer Scheduling of Public Transport: Urban Passenger Vehicle and Crew Scheduling*, pages 269–280. North-Holland, 1981.

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

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

[40] M. W. P. Savelsbergh. Preprocessing and probing techniques for mixed integer programming problems. *ORSA J. Comput.*, 6(4):445–454, 1994.

[41] SCIP. http://scip.zib.de, Solving constraint integer programs.

[42] J. I. A. Stallaert. The complementary class of generalized flow cover inequalities. *Discrete Applied Mathematics*, 77(1):73–80, 1997.

[43] X. Sun, X. Zheng, and D. Li. Recent advances in mathematical programming with semi-continuous variables and cardinality constraint. *Journal of the Operations Research Society of China*, 1:55–77, 2013.

[44] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83, 1945.

[45] L. A. Wolsey. Valid inequalities for 0-1 knapsacks and MIPs with generalised upper bound constraints. *Discrete Applied Mathematics*, 29(2–3):251–261, 1990.