

Monoidal Strengthening of Simple \mathcal{V} -Polyhedral Disjunctive Cuts

Aleksandr M. Kazachkov* Egon Balas†

February 28, 2023

Abstract

Disjunctive cutting planes can tighten a relaxation of a mixed-integer linear program. Traditionally, such cuts are obtained by solving a higher-dimensional linear program, whose additional variables cause the procedure to be computationally prohibitive. Adopting a \mathcal{V} -polyhedral perspective is a practical alternative that enables the separation of disjunctive cuts via a linear program with only as many variables as the original problem. The drawback is that the classical approach of monoidal strengthening cannot be directly employed without the values of the extra variables appearing in the extended formulation. We derive how to compute these values from a solution to the linear program generating \mathcal{V} -polyhedral disjunctive cuts. We then present computational experiments with monoidal strengthening of cuts from disjunctions with as many as 64 terms. Some instances are dramatically impacted, with strengthening increasing the gap closed by the cuts from 0 to 100%. However, for larger disjunctions, monoidal strengthening appears to be less effective, for which we identify a potential cause.

1 Introduction

Disjunction-based cutting planes, or *disjunctive cuts*, are a strong class of valid inequalities for mixed-integer programming problems, which can be used as a framework for analyzing or generating general-purpose cuts [8]. Their strength comes at a high computational cost, due to which only very special cases of disjunctive cuts have been deployed in optimization solvers. As a step towards practicality, Balas and Kazachkov [10] introduce a relaxation-based \mathcal{V} -polyhedral paradigm for disjunctive cuts, which trades off some theoretical strength for computational efficiency. The approach selects a small number of points and rays whose convex hull forms a relaxation of the disjunction; as a result, some potential cuts are no longer valid, but strong cuts are nevertheless guaranteed to be obtainable. Further, cuts

*University of Florida, Gainesville, FL, USA (akazachkov@ufl.edu).

†E. Balas passed away during the preparation of this manuscript, which started when both authors were at Carnegie Mellon University. The core ideas and early results are documented in the PhD dissertation of Kazachkov [37, Chapter 5]. A.M. Kazachkov completed the computational experiments, analysis, and writing independently.

from this relaxation, called \mathcal{V} -polyhedral (*disjunctive*) cuts (VPCs), can be generated via a relatively compact linear program, called the *point-ray linear program* (PRLP), compared to the usual higher-dimensional *cut-generating linear program* (CGLP) for disjunctive cuts [8, 14, 15]. Hence, with VPCs, it is more computationally efficient to improve the disjunction by adding terms and increase the relaxation quality, thereby accessing disjunctive cuts that differ substantially from the families of cuts typically applied in solvers.

VPCs improve the average (*integrality*) *gap closed* substantially relative to *Gomory mixed-integer cuts* (GMICs) and other standard cuts in solvers. However, the computational experiments by Balas and Kazachkov [10] reveal a curiosity: there are instances for which GMICs (which can be derived as cuts from a two-term disjunction) remain stronger than VPCs even when using large variable disjunctions. For example, for the instance `10teams`, originally part of the 3rd Mixed Integer Programming Library (MIPLIB) [18], GMICs close 100% of the integrality gap, while VPCs from a 64-term disjunction close 0% of the gap.

A potential explanation for this phenomenon is that GMICs benefit from a *strengthening* procedure that cannot be directly applied to VPCs. Specifically, the GMIC two-term disjunction can be obtained via *monoidal strengthening* of a disjunction on a single variable [9, 12, 38]. Monoidal strengthening of cuts from more general disjunctions is also possible, but the procedure ostensibly requires a *simple* disjunction, where each term only imposes a single new constraint. This is not a theoretical barrier, as any cut from a general disjunction can also be derived from a simple disjunction obtained from the general one by aggregating the constraints defining each disjunctive term. The multipliers for this aggregation are precisely the Farkas certificate for the validity of the cut. The key challenge for VPCs is that this certificate is not readily available, because the PRLP only has variables for the cut coefficients, compared to the CGLP that explicitly includes variables for the Farkas multipliers. Our contributions, summarized next, are to identify a way to efficiently apply monoidal strengthening for the particular version of the VPC framework introduced in Balas and Kazachkov [10], as well as to implement and computationally evaluate this strengthening idea.

Contributions. Given a VPC, one can solve the CGLP with cut coefficients fixed and retrieve the required values of the aggregation multipliers, in order to apply monoidal strengthening. Unfortunately, the computational effort associated to this is likely to be prohibitive. Our first contribution, discussed in [Section 3](#), is observing that solving the CGLP is unnecessary: it suffices to use the inverse of an easily-identified nonsingular matrix per disjunctive term. Furthermore, for the type of *simple* VPCs proposed and tested by Balas and Kazachkov [10], this inverse is readily available within the cut generation process.

Next, in [Section 4](#), we discuss computational experiments with strengthening simple VPCs on a set of benchmark instances. We compare the strength to unstrengthened VPCs and to GMICs, for disjunctions ranging in size up to 64 terms. We find that strengthening can significantly improve the gap closed for some instances. Furthermore, we see that GMICs and unstrengthened VPCs tend to be complementary in terms of which instances they benefit, but applying monoidal strengthening enables the two families to be simultaneously effective for more instances. The results are most striking for two-term disjunctions, in which strengthened VPCs close 40% more gap than unstrengthened VPCs, on average.

For example, returning to the instance `10teams`, the VPCs from a single variable disjunction close 0% of the integrality gap, but this value goes to 100% after strengthening the cuts. However, as the size of the disjunction increases, the relative improvement by strengthening becomes smaller. Our final contribution, in [Section 5](#), is identifying a theoretical source of this weakness.

Related Work. A focal point in the literature on monoidal strengthening for disjunctive cuts [9] (see also Balas [8, Section 7]) is the special case of *split disjunctions*, which are *parallel* two-term disjunctions that are used for GMICs and related cut families. In this context, the use of the CGLP leads to *lift-and-project cuts* (L&PCs) [14], to which monoidal strengthening can be applied [15, Section 2.4]. The family of strengthened L&PCs is equivalent to GMICs, as shown by Balas and Perregaard [12], and to mixed-integer rounding inequalities [45, 46], as discussed in Cornuéjols and Li [23]. Balas and Perregaard [12] provide an appealing geometric interpretation of this connection via intersection cuts [7]: every undominated L&PC can be derived as an intersection cut from a basis in the original problem space. As a result, L&PCs can be generated without explicitly building the CGLP and without hindering a posteriori strengthening of the cuts. Bonami [19] presents a different method for separating L&PCs in the original space of variables that is also amenable to strengthening. Avoiding formulating the higher-dimensional CGLP is the key advance that has enabled the effective inclusion of L&PCs in several solvers.

Sidestepping the CGLP continues to be crucial to move beyond split disjunctions. However, the aforementioned approaches [12, 19] rely on properties of the split set; for example, with general disjunctions, there exist cuts that dominate all intersection cuts [5, 40, 11], so one cannot hope to merely pivot among bases in the original space. Nonetheless, a stream of work [36, 20, 40] extends cut generation in the original space to general two-term disjunctions, and monoidal strengthening applies to the resulting cuts [28]. No further extension of this technique to more general disjunctions has been reported in the literature.

This motivates the use of VPCs, due to the PRLP’s advantage of having the same number of variables as the original problem. The difficulty is that a description of a polyhedron using points and rays may be exponentially larger than using inequalities, causing exponentially many constraints in the PRLP. This naturally leads to row generation in prior work by Perregaard and Balas [48] and Louveaux et al. [44] when invoking the \mathcal{V} -polyhedral perspective. In the experiments by Perregaard and Balas [48], for disjunctions with 16 terms, separating cuts via the PRLP with row generation is an order of magnitude faster than via the CGLP. Nonetheless, row generation is time consuming, as multiple PRLPs must be solved to find one valid inequality.

The remedy by Balas and Kazachkov [10] is to construct a relaxation of each disjunctive term, where the resulting PRLP has few rows and immediately produces valid cuts. This is successful at quickly generating cuts from large disjunctions, but the average gap closed by the cuts alone is less than that from GMICs. It is only when VPCs and GMICs are used together that a marked improvement in gap closed is observed, which shows that VPCs affect a different region of the relaxation than GMICs. However, as mentioned with the `10teams` instance in which GMICs close all of the gap, while VPCs close none, the results also suggest that the absence of strengthening for VPCs is a significant deficiency.

As discussed, the vanilla monoidal strengthening presented by Balas and Jeroslow [9] does not directly apply to VPCs due to the lack of the values of the aggregation multipliers. Balas and Qualizza [13, Section 6] show that a cross-polytope disjunction, arising from using multiple rows of the simplex tableau, can be strengthened by *modularizing* the inequalities defining the disjunction, replacing the coefficients of integer-restricted nonbasic variables, and they prove the form of the optimal strengthening for the two-row case.

An alternative to monoidal strengthening is the group-theoretic approach [35, 32], equivalent to monoidal strengthening under some conditions. Specifically, “trivial lifting” has been applied to simple disjunctions [26, 24, 16, 25, 49]. Evaluating the trivial lifting is expensive in general [30], and it does not directly apply to arbitrary disjunctive cuts.

While this paper exclusively approaches disjunctive cut generation via the VPC framework, there exist other methods for producing strong disjunctive cuts without solving the higher-dimensional CGLP. Any such approach could potentially benefit from the efficient computation of a Farkas certificate. For example, a common technique in the literature is to use a disjunction to strengthen cuts via tilting, which has been applied to linear and nonlinear integer optimization problems [47, 39, 37, 42].

2 Notation and Background

Our target is to find strong valid cuts to tighten the natural linear relaxation of the mixed-integer linear program below, given rational data:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^\top x \\ & A_i \cdot x \geq b_i \quad \text{for } i \in [q], \\ & x_j \geq 0 \quad \text{for } j \in [n], \\ & x_j \in \mathbb{Z} \quad \text{for } j \in \mathcal{I}. \end{aligned} \tag{IP}$$

Here, $[n] := \{1, \dots, n\}$ for any integer n , and $\mathcal{I} \subseteq [n]$ is the set of integer-restricted variables. For a given matrix A , we denote the i th row by “ A_i .” and the j th column by “ $A_{\cdot j}$.” Let P_I denote the feasible region of (IP), and let $P := \{x \in \mathbb{R}_{\geq 0}^n : Ax \geq b\}$.

One way to strengthen the formulation P (with respect to P_I) is to use logical conditions to formulate a *disjunction*, from which valid inequalities for P_I can then be derived. Suppose $\bigvee_{t \in \mathcal{T}} (D^t x \geq D_0^t)$ is a valid disjunction, in the sense that $P_I \subseteq \bigcup_{t \in \mathcal{T}} \{x \in \mathbb{R}^n : D^t x \geq D_0^t\}$. Let $Q^t := \{x \in P : D^t x \geq D_0^t\}$. This is an \mathcal{H} -polyhedral (inequality) description. We assume $Q^t \neq \emptyset$ for all $t \in \mathcal{T}$.

Let $P^t := \{x \in \mathbb{R}^n : A^t x \geq b^t\}$ denote a relaxation of Q^t , where $A^t x \geq b^t$ is defined by a subset of the constraints defining Q^t . For the VPC procedure, we must ensure that P^t has relatively few extreme points and rays, i.e., it has a compact \mathcal{V} -polyhedral description $(\mathcal{P}^t, \mathcal{R}^t)$, so that $P^t = \text{conv}(\mathcal{P}^t) + \text{cone}(\mathcal{R}^t)$. Define the *disjunctive hull* $P_D := \text{cl conv}(\bigcup_{t \in \mathcal{T}} P^t)$, which can be described by the point-ray collection $(\mathcal{P}, \mathcal{R}) := (\bigcup_{t \in \mathcal{T}} \mathcal{P}^t, \bigcup_{t \in \mathcal{T}} \mathcal{R}^t)$. For $t \in \mathcal{T}$, let q'_t be the number of rows of A^t . We first summarize some important disjunctive programming concepts and the two cut-generating paradigms that we are relating.

CGLP. One way to generate valid cuts for P_D is through the CGLP, which is an application of *disjunctive programming duality* [8, Section 4]. Specifically, an inequality $\alpha^\top x \geq \beta$ is valid for P_D if and only if the inequality is valid for each P^t , $t \in \mathcal{T}$. Consequently, by Farkas's lemma [27], $\alpha^\top x \geq \beta$ is valid for P_D if and only if the following system is feasible, in variables $(\alpha, \beta, \{v^t\}_{t \in \mathcal{T}})$, where $v^t \in \mathbb{R}^{1 \times q_t}$ is a row vector of appropriate length for each $t \in \mathcal{T}$:

$$\left. \begin{array}{l} \alpha^\top = v^t A^t \\ \beta \leq v^t b^t \\ v^t \in \mathbb{R}_{\geq 0}^{q_t} \end{array} \right\} \text{ for all } t \in \mathcal{T}. \quad (1)$$

We refer to $\{v^t\}_{t \in \mathcal{T}}$ as the *Farkas certificate* for the validity of $\alpha^\top x \geq \beta$ for P_D .

To generate cuts with (1), one typically maximizes the violation with respect to a P_I -infeasible point, after adding a normalization, which can be a crucial choice [29]. For example, the constant of the cut can be fixed to $\bar{\beta} \in \mathbb{R}$:

$$\{(\alpha, \{v^t\}_{t \in \mathcal{T}}) : (\alpha, \bar{\beta}, \{v^t\}_{t \in \mathcal{T}}) \text{ is feasible to (1)}\}. \quad (\text{CGLP}(\bar{\beta}))$$

PRLP. An alternative way to generate disjunctive cuts is through the *reverse polar* of P_D [8, Section 5], which is defined with respect to a given $\bar{\beta} \in \mathbb{R}$ as

$$\{\alpha \in \mathbb{R}^n : \alpha^\top x \geq \bar{\beta} \text{ for all } x \in P_D\}.$$

Clearly this captures all of the valid inequalities for P_D whose constant is equal to $\bar{\beta}$. Since $x \in P_D$ if and only if $x \in \text{conv}(\mathcal{P}) + \text{cone}(\mathcal{R})$, it holds that $\alpha^\top x \geq \bar{\beta}$ is valid for P_D if and only if it is satisfied by all of the points and rays in $(\mathcal{P}, \mathcal{R})$. This yields the system $(\text{PRLP}(\bar{\beta}))$, in variables $\alpha \in \mathbb{R}^n$, for a fixed $\bar{\beta}$:

$$\begin{array}{ll} \alpha^\top p \geq \bar{\beta} & \text{for all } p \in \mathcal{P} \\ \alpha^\top r \geq 0 & \text{for all } r \in \mathcal{R}. \end{array} \quad (\text{PRLP}(\bar{\beta}))$$

The feasible solutions to $(\text{PRLP}(\bar{\beta}))$ are what we refer to as VPCs.

As discussed, the advantage of $(\text{PRLP}(\bar{\beta}))$ over $(\text{CGLP}(\bar{\beta}))$ is the absence of the Farkas multipliers as variables, so VPCs are generated without requiring a lifted space. As we see next, the disadvantage to $(\text{PRLP}(\bar{\beta}))$ is that these missing variables are used in strengthening the cuts after they are generated.

Monoidal strengthening. Balas and Jeroslow [9] strengthen cuts with a *monoid*:

$$\mathbb{M} := \left\{ \mathbf{m} \in \mathbb{Z}^{|\mathcal{T}|} : \sum_{t \in \mathcal{T}} \mathbf{m}_t \geq 0 \right\}. \quad (\mathbb{M})$$

It is also assumed that, for each $t \in \mathcal{T}$, there exists a finite lower bound vector ℓ^t such that $D^t x \geq \ell^t$ for all $x \in P_I$. Let $\Delta^t := D_0^t - \ell^t$.

To strengthen the cut, we improve the underlying disjunction. Specifically, given a valid disjunction $\vee_{t \in \mathcal{T}} (D^t x \geq D_0^t)$, for any $\mathbf{m} \in \mathbb{M}$ and $k \in \mathcal{I}$, the disjunction $\vee_{t \in \mathcal{T}} (\tilde{D}^t x \geq \tilde{D}_0^t)$ is

also valid, where $\tilde{D}_{\cdot k}^t := D_{\cdot k}^t + \Delta^t \mathbf{m}_t$, and $\tilde{D}_{\cdot j}^t = D_{\cdot j}^t$ for all $j \neq k$. The strengthened cut is obtained by applying the Farkas certificate of the unstrengthened cut to the strengthened disjunction.

Let q_t denote the number of constraints in $D^t x \geq D_0^t$ for term $t \in \mathcal{T}$. Given row vectors $(u^t, u_0^t) \in \mathbb{R}_{\geq 0}^{1 \times q} \times \mathbb{R}_{\geq 0}^{1 \times q_t}$, define

$$\alpha_k^t := u^t A_{\cdot k} + u_0^t D_{\cdot k}^t. \quad (\alpha_k^t)$$

Then (using an appropriate CGLP) the cut $\alpha^\top x \geq \beta$ is valid for P_D , where

$$\alpha_k := \max_{t \in \mathcal{T}} \{\alpha_k^t\} \quad \text{and} \quad \beta := \min_{t \in \mathcal{T}} \{u^t b + u_0^t D_0^t\}.$$

(The above applies to cuts valid for $\bigvee_{t \in \mathcal{T}} Q^t$; for P_D , assume a value of zero for the multipliers on constraints of Q^t that are not present in P^t .) Define $\hat{u}_k^t := \alpha_k - \alpha_k^t$. We now apply monoidal strengthening to the cut $\alpha^\top x \geq \beta$.

Theorem 1 ([9, Theorem 3]). *Given $(u^t, u_0^t) \in \mathbb{R}_{\geq 0}^{1 \times q} \times \mathbb{R}_{\geq 0}^{1 \times q_t}$ for $t \in \mathcal{T}$, the inequality $\tilde{\alpha}^\top x \geq \beta$ is valid for P_I , where $\tilde{\alpha}_k := \alpha_k$ for $k \notin \mathcal{I}$, and, for $k \in \mathcal{I}$,*

$$\tilde{\alpha}_k := \inf_{m \in \mathbf{M}} \max_{t \in \mathcal{T}} \{\alpha_k^t + u_0^t \Delta^t \mathbf{m}_t\} = \alpha_k + \inf_{m \in \mathbf{M}} \max_{t \in \mathcal{T}} \{-\hat{u}_k^t + u_0^t \Delta^t \mathbf{m}_t\}.$$

Thus, the Farkas certificate $\{(u^t, u_0^t)\}_{t \in \mathcal{T}}$ is used for monoidal strengthening. Computing these values without solving the CGLP is our next target.

3 Correspondence Between PRLP and CGLP Solutions

Let $\bar{\alpha}^\top x \geq \bar{\beta}$ be a valid inequality for P_D , corresponding to a feasible solution to (PRLP($\bar{\beta}$)). Our goal is to compute Farkas multipliers certifying the cut's validity without explicitly solving the CGLP. While one can solve for values v^t that satisfy $\bar{\alpha}^\top = v^t A^t$, $\bar{\beta} = v^t b^t$, $v^t \geq 0$, we provide an improvement via basic linear programming concepts. We first present a special case in Sections 3.1 and 3.2, when the disjunctive terms P^t are not primal degenerate, a condition that is satisfied by the VPC procedure implemented for our experiments. Then, Section 3.3 discusses a challenge posed by the general case.

We assume that $\bar{\alpha}^\top x \geq \bar{\beta}$ is supporting for all terms in \mathcal{T} . This is for ease of notation, as otherwise we would need to add an index t to the constant side. Concretely, the assumption is without loss of generality because, for any term $t \in \mathcal{T}$, we can increase the constant side of the cut until we obtain an inequality $\bar{\alpha}^\top x \geq \bar{\beta}_t$ that is supporting for term t , though perhaps invalid for other terms. The value of $\bar{\beta}_t$ can be quickly calculated by taking the dot product of $\bar{\alpha}$ with every point in \mathcal{P}^t . We can then find a certificate v^t of the validity of $\bar{\alpha}^\top x \geq \bar{\beta}_t$ for P^t , which also serves as a certificate for the weaker inequality $\bar{\alpha}^\top x \geq \bar{\beta}$. We state, without proof, a slightly more general version of this in Lemma 2.

Lemma 2. *For $t \in \mathcal{T}$, let $C^t \supseteq P^t$ and $\bar{\beta}_t \geq \bar{\beta}$ such that $\bar{\alpha}^\top x \geq \bar{\beta}_t$ is valid for C^t . Then, given any Farkas certificate for the validity of the inequality $\bar{\alpha}^\top x \geq \bar{\beta}_t$ for C^t , the same multipliers certify that $\bar{\alpha}^\top x \geq \bar{\beta}$ is valid for P^t . \square*

For convenience, we introduce extra notation to refer to the feasible region of Q^t as $\hat{A}^t x \geq \hat{b}^t$, and we define the number of these constraints as $\hat{q}_t := q + q_t + n$. For $N \subseteq [\hat{q}_t]$, define $\hat{A}_{N^t}^t x \geq \hat{b}_{N^t}^t$ as the constraints of Q^t indexed by N .

3.1 Simple VPCs

Our experimental setup in [Section 4](#) follows that of Balas and Kazachkov [10], who focus on a variant of the VPC framework called *simple* VPCs. Let p^t be a vertex of Q^t , for $t \in \mathcal{T}$. There exists a *cobasis* for p^t , a set of n linearly independent constraints among those defining Q^t that are tight at p^t . Let $N^t \subseteq [\hat{q}_t]$ denote the indices of these n constraints, and define the *basis cone* $C^t := \{x \in \mathbb{R}^n : \hat{A}_{N^t}^t x \geq \hat{b}_{N^t}^t\}$. The inequality $\bar{\alpha}^\top x \geq \bar{\beta}$ is a simple VPC if P^t is a basis cone for each term. The (translated) cone C^t has a particularly easy \mathcal{V} -polyhedral representation: there is a single extreme point p^t , and there are n extreme rays $\{r^i\}_{i \in [n]}$. The i th extreme ray of C^t corresponds to increasing the “slack” on the i th constraint defining C^t [21, Chapter 6]. [Lemma 3](#) states that, for simple VPCs, the values of the variables $\{v^t\}_{t \in \mathcal{T}}$ to $(\text{CGLP}(\bar{\beta}))$ can be computed via the dot product of the cut coefficients with the rays of C^t .

Lemma 3. *Let C^t be a basis cone defined by N^t , the indices of n linearly independent constraints of Q^t . If $\bar{\alpha}^\top x \geq \bar{\beta}$ is valid for C^t , then the multiplier on constraint $i \in [n]$ of C^t has value $v_i^t = \bar{\alpha}^\top r^i$, where r^i is column i of $(\hat{A}_{N^t}^t)^{-1}$.*

Proof. Add nonnegative slack variables $s_{N^t}^t$ for each row indexed by N^t , so that $\hat{A}_{N^t}^t x - s_{N^t}^t = b_{N^t}^t$. Then observe that, being a cobasis, $\hat{A}_{N^t}^t$ is invertible, so $x = (\hat{A}_{N^t}^t)^{-1} b_{N^t}^t + (\hat{A}_{N^t}^t)^{-1} s_{N^t}^t = p^t + \sum_{i \in N^t} r^i s_i^t$. The last equality follows from the derivation of the rays of C^t ; see, for example, Conforti et al. [21, Chapter 6]. \square

Therefore, for simple VPCs, the Farkas certificate can be computed with no extra effort when given the point-ray representation of P_D . Moreover, Balas and Kazachkov [10] obtain simple VPCs from the leaf nodes of a partial branch-and-bound tree and use p^t as the optimal solution to the linear relaxation at each leaf; implemented carefully, this can further reduce the computational load for generating then strengthening VPCs, as the values of the rays can be read from the optimal tableau, which is typically readily available from a solver.

3.2 Relaxations Without Primal Degeneracy

Suppose the relaxation $P^t \supseteq Q^t$ is a *simple polyhedron*, in which every extreme point and ray is defined by a *unique* basis [50]. The basis cone C^t used for simple VPCs is one example. While the basis cone setting may seem quite narrow, it turns out to encompass more general situations. Specifically, there always exists a basis cone $C^t \supseteq P^t$ such that $\bar{\alpha}^\top x \geq \bar{\beta}$ is valid and supporting for C^t .

Lemma 4. *Let P^t be a simple polyhedron, and suppose the point-ray collection $(\mathcal{P}^t, \mathcal{R}^t)$ satisfies $P^t = \text{conv}(\mathcal{P}^t) + \text{cone}(\mathcal{R}^t)$. Let $\bar{\alpha}^\top x \geq \bar{\beta}$ be a valid inequality for P^t . Then there exists a vertex $p^t \in \mathcal{P}^t$ such that $\bar{\alpha}^\top x \geq \bar{\beta}$ is valid for the basis cone C^t associated to p^t , defined with respect to the constraints of P^t .*

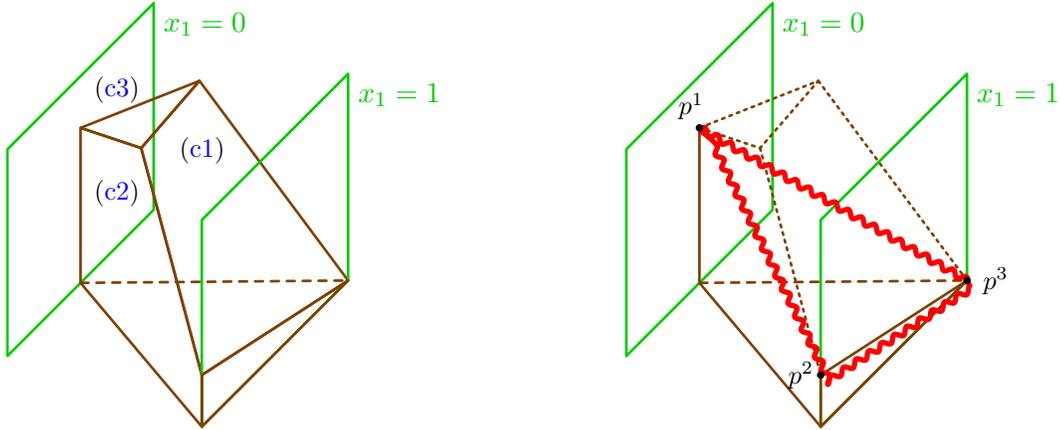


Figure 1: [Example 1](#): Disjunctive terms with primal degeneracy, despite a nondegenerate initial polyhedron. The VPC is the red wavy line in the second panel.

Proof. Let p^t be an optimal solution to $\min_x \{\bar{\alpha}^\top x : x \in P^t\} = \min_p \{\bar{\alpha}^\top p : p \in \mathcal{P}^t\}$. Define $\bar{\beta}_t := \bar{\alpha}^\top p^t$. Note that the rays in \mathcal{R}^t need not be considered, as the optimization problem must be bounded since $\bar{\alpha}^\top x \geq \bar{\beta}_t$ is valid for all $x \in P^t$. The point p^t has a unique basis, so the basis cone C^t is defined by the (precisely) n constraints of P^t that are tight at p^t . Optimality of p^t implies all reduced costs are nonnegative. It follows that $\bar{\alpha}^\top r \geq 0$ every ray $r \in C^t$. Since $\bar{\alpha}^\top p^t = \bar{\beta}_t \geq \bar{\beta}$, the inequality $\bar{\alpha}^\top x \geq \bar{\beta}$ is valid for C^t . \square

Therefore, we can invoke [Lemmas 2](#) and [3](#) to find the Farkas certificate for this case. Note that, when the given point-ray collection only contains extreme points and rays, the rays of C^t for any basis cone of the simple polyhedron P^t can be computed as the rays \mathcal{R}^t , along with the directions $p - p^t$ for every point $p \in \mathcal{P}^t$ that is adjacent (one pivot away) from p^t .

3.3 Relaxations with Primal Degeneracy

Up to now, we have made the convenient assumption that the relaxation P^t is a simple polyhedron. More generally, there always exists a basis cone C^t , such that a cut valid for P^t is valid for C^t . With [Example 1](#), we illustrate the complication if $\bar{\alpha}^\top x \geq \bar{\beta}$ is supporting at a primal degenerate point of P^t : a basis for that point needs to be chosen carefully, as the inequality may not be valid for some basis cones. It can be computationally involved to find a valid basis in these situations, which prevents a direct application of our approach relying on simple polyhedra. The purpose of this example is to highlight a crucial obstacle to a complete correspondence between PRLP and CGLP solutions, but we do not further investigate the nondegenerate case in this paper.

Example 1. Figure 1 shows a polyhedron P , defined as the feasible solutions to

$$-(13/8)x_1 - (1/4)x_2 - x_3 \geq -15/8 \tag{c1}$$

$$(1/2)x_1 + x_2 \geq 1/2 \tag{c2}$$

$$(1/2)x_1 - x_3 \geq -3/4 \tag{c3}$$

$$(1/2)x_1 - x_2 \geq -1/2 \tag{c4}$$

$$x_2 \geq 0. \tag{c5}$$

A valid cut from the disjunction $(-x_1 \geq 0) \vee (x_1 \geq 1)$ has coefficients $\bar{\alpha}^\top = (-5/8, -1/4, -1)$ and constant $\bar{\beta} = -7/8$. The cut, depicted in the right panel, is incident to point $p^1 = (0, 1/2, 3/4)$ on $P^1 := \{x \in P : -x_1 \geq 0\}$. This point is tight for four inequalities: three defining P (constraints (c2)–(c4)), and the disjunction-defining inequality $-x_1 \geq 0$. Note that P is simple, but P^1 is not.

To construct the cobasis N^1 , such that the inequality is valid for the associated basis cone C^1 , we must select three linearly independent constraints among those that are tight at p^1 . One of the inequalities must be $-x_1 \geq 0$, as otherwise we have not imposed the disjunction at all (but we also know the cut is not valid for P). It can be verified that the only valid choice for this example is N^1 containing the indices for (c3), (c4), and the disjunctive inequality $-x_1 \geq 0$. ■

4 Computational Experiments

We implement monoidal strengthening for simple VPCs, building on the code used by Balas and Kazachkov [10] from <https://github.com/akazachk/vpc>. Our goal for the computational study is to measure the effect of monoidal strengthening on the *percent integrality gap closed* by VPCs, compared to unstrengthened VPCs and GMICs, and evaluated across different disjunction sizes.

The code is run on HiPerGator, a shared cluster through Research Computing at the University of Florida. The computational setup is nearly identical to the one described in Balas and Kazachkov [10, Section 5 and Appendix C]. We select instances from the union of the MIPLIB [17, 18, 4, 41, 31], CORAL [22], and NEOS sets, restricted to those with at most 5,000 rows and columns and based on other criteria given in [10, Appendix C]. This yields 332 instances suitable for gap closed comparisons. However, we only report on 274 of these 332 instances, due to memory resource constraints on the cluster. Despite this reduced dataset, we can identify recurring patterns in how monoidal strengthening affects instances. Instances are presolved with Gurobi [34], but cut generation is done via the C++ interface to COIN-OR [43], using Clp [3] for solving linear programs and Cbc [1] for constructing disjunctions based on partial branch-and-bound trees. We test six different disjunction sizes, stopping branching when the number of leaf nodes (disjunctive terms) is 2^ℓ for $\ell \in [6]$. Thus, we report results with monoidal strengthening of disjunctive cuts from up to 64-term disjunctions, though only one disjunction is used at a time. One GMIC is generated per fractional integer variable at an optimal solution to the linear programming relaxation, and the number of GMICs is also used as the limit for the number of VPCs we generate for that instance per fixed choice of disjunction. One round of cuts is used for both

procedures. GMICs are generated through `Cg1GMI` [2], while the VPC generation procedure is identical to that of Balas and Kazachkov [10], with strengthening applied afterwards.

While Lemma 3 enables us to calculate the values of the Farkas multipliers via the rays of each relaxation P^t , and these values are readily available based on how we built the PRLP, we do not avail of this connection. Instead, we calculate $v^t = \bar{\alpha}^\top (A^t)^{-1}$. This approach is still more direct than solving a feasibility version of $(\text{CGLP}(\bar{\beta}))$ with $\bar{\alpha}$ fixed. We opt for numerical safety for this exploratory investigation, so we use the Eigen library [33] to recompute the inverse of A^t rather than reading from the Cbc / Clp internal basis inverse for each term.

We report the average percent integrality gap closed by VPCs and GMICs in Table 1. The first six data rows contain the results for each fixed disjunction size. The penultimate data row, labeled “Best”, uses the highest gap closed per instance across all disjunctions. The last data row, labeled “Wins”, reports the number of instances for which the “Best” gap closed is at least 10^{-3} higher than the gap closed by GMICs. In the columns, we refer to GMICs by “G”, unstrengthened VPCs by “V”, strengthened VPCs by “V+”. The columns “G+V” and “G+V+” refer to GMICs applied together with VPCs. There are two sets of instances: “All” reports on all 274 instances, while “ $\geq 10\%$ ” reports on the 97 instances for which unstrengthened VPCs alone close at least 10% of the integrality gap for the “Best” values.

In terms of overall gap closed, despite the monoidal strengthening procedure, as reported by Balas and Kazachkov [10], VPCs alone do not outperform GMICs for the “All” set, but using VPCs and GMICs together provides around 40% improvement in gap closed relative to GMICs alone. While adding VPCs with GMICs might double the number of cuts, one round of VPCs continues to close substantial more gap even after multiple rounds of solver-default cuts [10]. Hence, VPCs tighten the relaxation in different regions relative to GMICs. This is also highlighted by the “ $\geq 10\%$ ” set, which are instances for which VPCs have strong performance; for this set, GMICs are relatively weaker, with the best VPCs per instance (used alone) providing a 75% improvement in average percent gap closed over GMICs alone. We also see this in the “Wins” row: for the “ $\geq 10\%$ ” set, VPCs alone outperform GMICs for 73 of the 97 instances in the set.

Next, we summarize observations about the effect of monoidal strengthening. We start with the first data row, in which VPCs are derived from one split disjunction per instance. For the set “All”, monoidal strengthening affects the gap closed by VPCs for 87 instances and increases the average gap closed by VPCs by $\sim 1\%$ from 2.28% to 3.25%, a 40% relative improvement. For the set “ $\geq 10\%$ ”, the corresponding relative improvement is 20%.

Although the two-term case is encouraging, and a similar relative improvement in gap closed would be substantial for larger disjunctions, this unfortunately does not materialize. From Table 1, we see that as the disjunction size increases, the contribution of monoidal strengthening tends to further diminish, with an absolute improvement in gap closed of only 0.1% for VPCs from a 64-term disjunction. We will discuss a potential cause for this in the next section.

We now compare the columns “G+V+” to “G+V”. On the set “All”, even for split disjunctions, the effect of strengthening is minimal when VPCs are combined with GMICs, with strengthening only yielding an additional 0.23% in percent gap closed, preserving around 23% of the improvement between “V+” and “V”. For larger disjunctions, while the absolute

Table 1: Average percent gap closed by VPCs and GMICs according to the number of leaf nodes used to construct the partial branch-and-bound tree. “Best” refers to the maximum gap closed per instance across all partial tree sizes.

	All					$\geq 10\%$				
	G	V	V ⁺	G+V	G+V ⁺	G	V	V ⁺	G+V	G+V ⁺
2 leaves	17.21	2.28	3.25	17.95	18.18	16.29	5.34	6.47	18.13	18.59
4 leaves	17.21	3.35	3.72	18.37	18.54	16.29	7.81	8.35	19.14	19.48
8 leaves	17.21	4.51	4.76	18.98	19.15	16.29	10.84	11.16	20.66	20.91
16 leaves	17.21	6.41	6.57	20.54	20.67	16.29	15.81	16.05	24.86	25.04
32 leaves	17.21	8.78	8.97	22.31	22.48	16.29	21.82	22.28	29.59	29.97
64 leaves	17.21	10.46	10.57	23.72	23.83	16.29	25.59	25.85	32.90	33.14
Best	17.21	11.93	12.57	24.67	24.89	16.29	29.26	29.53	35.27	35.59
Wins		103	104	185	190		73	73	94	94

increase in gap closed by strengthened VPCs is small, over 80% of that improvement is preserved when adding GMICs together with VPCs.

A closer examination of the results supports the hypothesis that monoidal strengthening is a key factor enabling GMICs to close more gap than VPCs. We sort the instances by the increase in gap closed by strengthened VPCs compared to unstrengthened ones, using the best gap closed across all disjunction sizes, per column. Table 2 shows the top ten instances, sorted by the last column, which calculates the difference between “V⁺” and “V”. The table includes the instance `10teams` discussed earlier, as well as six other instances for which unstrengthened VPCs close at most 5% of the gap. We see that monoidal strengthening of VPCs bridges a large portion of the difference with GMICs for these instances. For `neos-1281048`, the situation is reversed: 121 GMICs close no gap while 29 unstrengthened VPCs close 17% of the gap, which is further improved to 29% after strengthening. From this table, we also observe the phenomenon that the value in column “G+V” is typically either entirely due to GMICs or to VPCs, but which cuts are more important varies by instance. The situation remains similar for the column “G+V⁺”, though now we find several cases (`f2gap401600`, `p0548`, `mkc`) in which the two cut families add to each other.

While running time is not our focus, and the shared computing environment makes wall clock times unreliable, Table 3 provides the average number of seconds for a single run of each instance, including generating then strengthening VPCs. On average, cut generation takes, in total, from less than a second for two-term disjunctions to 50 seconds for 16-term disjunctions, 150 seconds for 32-term disjunctions, and nearly 9 minutes for 64-term disjunctions. The time per cut, on average, is less than 0.1 seconds for two-term disjunctions, ranging up to 9 seconds for 32 terms and over 30 seconds for 64 terms.

Table 2: Percent gap closed for instances where strengthening VPCs works best.

Instance	G	V	V ⁺	G+V	G+V ⁺	V ⁺ -V
10teams	100.00	0.00	100.00	100.00	100.00	100.00
neos-1281048	0.00	17.09	29.36	17.09	29.36	12.27
neos-1599274	34.65	0.00	11.19	34.65	34.65	11.19
f2gap401600	62.97	2.53	11.34	63.31	71.77	8.80
prod2	2.31	27.60	35.90	27.63	35.91	8.29
neos-942830	6.25	0.00	6.25	6.25	6.25	6.25
p0548	48.62	3.28	9.03	49.03	55.11	5.75
mkc	6.08	2.60	6.56	6.35	9.61	3.96
f2gap201600	60.27	8.58	12.13	60.27	60.27	3.56
neos-4333596-skien	20.84	7.05	9.83	20.84	20.85	2.78

5 Choosing a Relaxation Amenable to Strengthening

In this section, we examine a potential cause of the diminishing effect of monoidal strengthening with larger disjunctions. From [Theorem 1](#), given an initial cut $\alpha^\top x \geq \beta$, we can strengthen coefficient α_k , $k \in \mathcal{I}$, to

$$\tilde{\alpha}_k = \alpha_k + \inf_{\mathbf{m} \in \mathbf{M}} \max_{t \in \mathcal{T}} \{-\hat{u}_k^t + u_0^t \Delta^t \mathbf{m}_t\},$$

where $\hat{u}_k^t = \alpha_k - (u^t A_{.k} + u_0^t D_{.k}^t)$ is the slack on the CGLP constraint $\alpha_k \geq u^t A_{.k} + u_0^t D_{.k}^t$. Equivalently, \hat{u}_k^t is the Farkas multiplier for the nonnegativity constraint $x_k \geq 0$. The next lemma restates the (known) reason that a *nonbasic* integral variable k is required for monoidal strengthening.

Lemma 5. *If $\hat{u}_k^t = 0$, then $\tilde{\alpha}_k = \alpha_k$.*

Proof. In this case, $\tilde{\alpha}_k = \alpha_k + \inf_{\mathbf{m} \in \mathbf{M}} \max_{t \in \mathcal{T}} \{u_0^t \Delta^t \mathbf{m}_t\}$. Since $\sum_{t \in \mathcal{T}} \mathbf{m}_t \geq 0$ for every $\mathbf{m} \in \mathbf{M}$, and $u_0^t \Delta^t \geq 0$, the optimal solution is $\mathbf{m} = \mathbf{0}$. \square

In the correspondence in [Section 3](#), we ultimately find a point $p^t \in P^t$ such that $\bar{\alpha}^\top p^t = \bar{\beta}_t$, where $\bar{\beta}_t = \min_p \{\bar{\alpha}^\top p : p \in \mathcal{P}^t\}$. We then compute a basis cone at p^t for which the cut is valid and use this (translated) cone to compute the values of the Farkas certificate. However, by complementary slackness, if $p_k^t > 0$, then necessarily $\hat{u}_k^t = 0$.

Table 3: Average time (seconds) to generate the cuts in column V⁺ of [Table 1](#).

Statistic	Set	2 leaves	4 leaves	8 leaves	16 leaves	32 leaves	64 leaves
Cut time (s)	All	0.76	6.39	15.33	49.90	149.84	525.78
	$\geq 10\%$	0.92	9.31	21.06	130.45	273.51	521.99
Time/cut (s)	All	0.08	0.39	0.97	2.65	9.00	30.54
	$\geq 10\%$	0.07	0.35	0.79	2.46	7.75	20.19

Although at first this appears simultaneously unfortunate and unavoidable, there are two potential remedies. First, there may be dual degeneracy in the choice of p^t : each such point can lead to a different Farkas certificate and therefore a different strengthening. Second, as observed by Balas and Qualizza [6], “sometimes weakening a disjunction helps the strengthening”. Though in that context, the weakening involves adding terms to the disjunction, the sentiment applies to our setting as well: if $\bar{\beta}_t > \bar{\beta}$, then one can seek a different, potentially infeasible, basis of Q^t in which more integer variables are nonbasic and $\bar{\alpha}^\top x \geq \bar{\beta}$ is still valid for the associated basis cone.

The computational results support the above intuition. When VPCs are generated from a split disjunction, on average, around 95% of the generated cuts per instance have any coefficient strengthened with the monoidal technique. This decreases to 85% for 64-term disjunctions. Furthermore, on average among VPCs to which strengthening has been applied, 20% of the cut coefficients are strengthened for split disjunctions, while this value steadily decreases as disjunction size increases, so among the analogous VPCs from 64-term disjunctions, only 10% of the coefficients are strengthened.

6 Conclusion

We show that strengthening cuts from general disjunctions is possible without explicitly solving a higher-dimensional CGLP, and that this strengthening can have a high impact for certain instances. However, several challenges are also highlighted for future work. First, the strengthening does not work well on average for larger disjunctions. While we propose a viable explanation and remedy, it is computationally demanding and requires development. Second, the optimal monoidal strengthening involves solving an integer program per cut; this is a relatively small and easy problem, but it nonetheless can be slow for larger disjunctions, as suggested by Table 3, which includes strengthening time. One can reduce this load by selectively strengthening only the most promising cuts, identified by theoretical properties or good heuristics, or to forego optimality in the strengthened cut coefficients. Our computational results indicate that VPCs and GMICs seem to have complementary affects; understanding this better is an opportunity to more widely adopt disjunctive cuts.

References

- [1] COIN-OR Branch and Cut. <https://github.com/coin-or/Cbc>.
- [2] COIN-OR Cut Generation Library. <https://github.com/coin-or/Cgl>.
- [3] COIN-OR Linear Programming. <https://github.com/coin-or/Clp>.
- [4] T. Achterberg, T. Koch, and A. Martin. MIPLIB 2003. *Oper. Res. Lett.*, 34(4):361–372, 2006.
- [5] Kent Andersen, Gérard Cornuéjols, and Yanjun Li. Split closure and intersection cuts. *Math. Program.*, 102(3, Ser. A):457–493, 2005.

- [6] E. Balas and A. Qualizza. Monoidal cut strengthening revisited. *Discrete Optim.*, 9(1): 40–49, 2012.
- [7] Egon Balas. Intersection cuts—a new type of cutting planes for integer programming. *Oper. Res.*, 19(1):19–39, 1971.
- [8] Egon Balas. Disjunctive programming. *Ann. Discrete Math.*, 5:3–51, 1979.
- [9] Egon Balas and Robert G. Jeroslow. Strengthening cuts for mixed integer programs. *European J. Oper. Res.*, 4(4):224–234, 1980.
- [10] Egon Balas and Aleksandr M. Kazachkov. \mathcal{V} -polyhedral disjunctive cuts, 2022. URL <https://arxiv.org/abs/2207.13619>.
- [11] Egon Balas and Tamás Kis. On the relationship between standard intersection cuts, lift-and-project cuts and generalized intersection cuts. *Math. Program.*, pages 1–30, 2016.
- [12] Egon Balas and Michael Perregaard. A precise correspondence between lift-and-project cuts, simple disjunctive cuts, and mixed integer Gomory cuts for 0-1 programming. *Math. Program.*, 94(2-3, Ser. B):221–245, 2003. The Aussois 2000 Workshop in Combinatorial Optimization.
- [13] Egon Balas and Andrea Qualizza. Intersection cuts from multiple rows: a disjunctive programming approach. *EURO J. Computat. Optim.*, 1(1):3–49, 2013.
- [14] Egon Balas, Sebastián Ceria, and Gérard Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Math. Program.*, 58(3, Ser. A):295–324, 1993.
- [15] Egon Balas, Sebastián Ceria, and Gérard Cornuéjols. Mixed 0-1 programming by lift-and-project in a branch-and-cut framework. *Man. Sci.*, 42(9):1229–1246, 1996.
- [16] Amitabh Basu, Pierre Bonami, Gérard Cornuéjols, and François Margot. Experiments with two-row cuts from degenerate tableaux. *INFORMS J. Comput.*, 23(4):578–590, 2011.
- [17] R. E. Bixby, E. A. Boyd, and R. R. Indovina. MIPLIB: A test set of mixed integer programming problems. *SIAM News*, 25:16, 1992.
- [18] R. E. Bixby, S. Ceria, C. M. McZeal, and M. W. P Savelsbergh. An updated mixed integer programming library: MIPLIB 3.0. *Optima*, 58:12–15, 6 1998.
- [19] Pierre Bonami. On optimizing over lift-and-project closures. *Math. Program. Comput.*, 4(2):151–179, 2012.
- [20] Pierre Bonami, Michele Conforti, Gérard Cornuéjols, Marco Molinaro, and Giacomo Zambelli. Cutting planes from two-term disjunctions. *Oper. Res. Lett.*, 41(5):442–444, 2013.

- [21] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. *Integer Programming*, volume 271 of *Graduate Texts in Mathematics*. Springer, Cham, 2014.
- [22] CORAL. Computational Optimization Research at Lehigh. MIP instances. coral.ise.lehigh.edu/data-sets/mixed-integer-instances/, 2020. Accessed September 2020.
- [23] Gérard Cornuéjols and Yanjun Li. Elementary closures for integer programs. *Oper. Res. Lett.*, 28(1):1–8, 2001.
- [24] Santanu S. Dey and Laurence A. Wolsey. Two row mixed-integer cuts via lifting. *Math. Program.*, 124(1-2, Ser. B):143–174, 2010.
- [25] Santanu S. Dey, Andrea Lodi, Andrea Tramontani, and Laurence A. Wolsey. On the practical strength of two-row tableau cuts. *INFORMS J. Comput.*, 26(2):222–237, 2014.
- [26] Daniel G. Espinoza. Computing with multi-row Gomory cuts. *Oper. Res. Lett.*, 38(2):115–120, 2010.
- [27] Julius Farkas. Theorie der einfachen Ungleichungen. *J. Reine Angew. Math.*, 124:1–27, 1902.
- [28] Tobias Fischer and Marc E. Pfetsch. Monoidal cut strengthening and generalized mixed-integer rounding for disjunctions and complementarity constraints. *Oper. Res. Lett.*, 45(6):556–560, 2017.
- [29] Matteo Fischetti, Andrea Lodi, and Andrea Tramontani. On the separation of disjunctive cuts. *Math. Program.*, 128(1-2, Ser. A):205–230, 2011.
- [30] Ricardo Fukasawa, Laurent Poirrier, and Álinson S. Xavier. The (not so) trivial lifting in two dimensions. *Math. Program. Comp.*, 11(2):211–235, 2019.
- [31] A. Gleixner, G. Hendel, G. Gamrath, T. Achterberg, M. Bastubbe, T. Berthold, et al. MIPLIB 2017: Data-Driven Compilation of the 6th Mixed-Integer Programming Library. *Math. Prog. Comp.*, 2021.
- [32] Ralph E. Gomory and Ellis L. Johnson. Some continuous functions related to corner polyhedra. *Math. Program.*, 3(1):23–85, 1972.
- [33] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. <http://eigen.tuxfamily.org>, 2010.
- [34] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2022.
- [35] Ellis L. Johnson. On the group problem for mixed integer programming. *Math. Program. Stud.*, (2):137–179, 1974.
- [36] Joaquim J. Júdice, Hanif D. Sherali, Isabel M. Ribeiro, and Ana M. Faustino. A complementarity-based partitioning and disjunctive cut algorithm for mathematical programming problems with equilibrium constraints. *J. Global Optim.*, 36(1):89–114, 2006.

- [37] Aleksandr M. Kazachkov. *Non-Recursive Cut Generation*. PhD thesis, Carnegie Mellon University, 2018.
- [38] Aleksandr M. Kazachkov and Felipe Serrano. Monoidal cut strengthening. In Oleg Prokopyev and Panos M. Pardalos, editors, *Encyclopedia of Optimization*. Springer US, Boston, MA. Under review.
- [39] Mustafa Kılınç, Jeff Linderoth, James Luedtke, and Andrew Miller. Strong-branching inequalities for convex mixed integer nonlinear programs. *Comput. Optim. Appl.*, 59(3): 639–665, 2014.
- [40] Tamás Kis. Lift-and-project for general two-term disjunctions. *Discrete Optim.*, 12: 98–114, 2014.
- [41] T. Koch, T. Achterberg, E. Andersen, O. Bastert, T. Berthold, R. E. Bixby, et al. MIPLIB 2010: mixed integer programming library version 5. *Math. Program. Comput.*, 3(2):103–163, 2011.
- [42] Jan Kronqvist and Ruth Misener. A disjunctive cut strengthening technique for convex MINLP. *Optim. Eng.*, 22(3):1315–1345, 2021.
- [43] Robin Lougee-Heimer. The Common Optimization INterface for Operations Research: Promoting open-source software in the operations research community. *IBM Journal of Research and Development*, 47, 2003.
- [44] Quentin Louveaux, Laurent Poirrier, and Domenico Salvagnin. The strength of multi-row models. *Math. Program. Comput.*, 7(2):113–148, 2015.
- [45] George L. Nemhauser and Laurence A. Wolsey. *Integer and combinatorial optimization*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Inc., New York, 1988.
- [46] George L. Nemhauser and Laurence A. Wolsey. A recursive procedure to generate all cuts for 0-1 mixed integer programs. *Math. Program.*, 46(1):379–390, 1990.
- [47] Michael Perregaard. *Generating Disjunctive Cuts for Mixed Integer Programs*. PhD thesis, Carnegie Mellon University, 9 2003.
- [48] Michael Perregaard and Egon Balas. Generating cuts from multiple-term disjunctions. In *Integer Programming and Combinatorial Optimization*, volume 2081 of *Lecture Notes in Comput. Sci.*, pages 348–360. Springer, Berlin, 2001.
- [49] Álinson S. Xavier, Ricardo Fukasawa, and Laurent Poirrier. Multirow intersection cuts based on the infinity norm. *INFORMS J. Comput.*, 33(4):1624–1643, 2021.
- [50] Günter M. Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1995.