SIAM J. OPTIM.
Vol. 31, No. 4, pp. 2539–2557




# LATTICE REFORMULATION CUTS*

KAREN AARDAL†, ANDREA LODI‡, ANDREA TRAMONTANI§, FREDERIK VON HEYMANN¶, AND LAURENCE A. WOLSEY‖

**Abstract.** Here we consider the question whether the lattice reformulation of a linear integer program can be used to produce effective cutting planes. In particular, we aim at deriving split cuts that cut off more of the integrality gap than Gomory mixed-integer (GMI) inequalities generated from LP-tableaus, while being less computationally demanding than generating the split closure. We consider integer programs (IPs) in the form $\max\{\boldsymbol{cx} \mid \boldsymbol{Ax} = \boldsymbol{b}, \boldsymbol{x} \in \mathbb{Z}^n_+\}$, where the reformulation takes the form $\max\{\boldsymbol{cx}^0 + \boldsymbol{cQ\mu} \mid \boldsymbol{Q\mu} \geq -\boldsymbol{x}^0,\ \boldsymbol{\mu} \in \mathbb{Z}^{n-m}\}$, where $\boldsymbol{Q}$ is an $n \times (n-m)$ integer matrix. Working on an optimal LP-tableau in the $\boldsymbol{\mu}$-space allows us to generate $n-m$ GMIs in addition to the $m$ GMIs associated with the optimal tableau in the $\boldsymbol{x}$ space. These provide new cuts that can be seen as GMIs associated to $n-m$ nonelementary split directions associated with the reformulation matrix $\boldsymbol{Q}$. On the other hand it turns out that the corner polyhedra associated to an LP basis and the GMI or split closures are the same whether working in the $\boldsymbol{x}$ or $\boldsymbol{\mu}$ spaces. Our theoretical derivations are accompanied by an illustrative computational study. The computations show that the effectiveness of the cuts generated by this approach depends on the quality of the reformulation obtained by the reduced basis algorithm used to generate $\boldsymbol{Q}$ and that it is worthwhile to generate several rounds of such cuts. However, the effectiveness of the cuts deteriorates as the number of constraints is increased.

**Key words.** integer programming, cutting planes, lattice reformulations, lattice basis reduction

**AMS subject classifications.** 90C10, 11H06

**DOI.** 10.1137/19M1291145

**1. Introduction.** In a series of papers Aardal et al. [3, 4, 1] have shown that certain integer programs that cannot be solved by a standard mixed integer programming (MIP) solver can be solved by using a lattice-reformulation of the problem. This raises the question studied here of whether such a lattice-reformulation can also be used to produce effective cutting planes.

Specifically we consider pure integer programs (IPs) in the form

$$\max\{\boldsymbol{cx} \mid \boldsymbol{Ax} = \boldsymbol{b}, \boldsymbol{x} \in \mathbb{Z}^n_+\}, \tag{1.1}$$

where $\boldsymbol{A} \in \mathbb{Z}^{m\times n}$, $\boldsymbol{b} \in \mathbb{Z}^m$, $\boldsymbol{c} \in \mathbb{Z}^n$ with $rank(\boldsymbol{A}) = m$ and $n > m$. Let $P = \{\boldsymbol{x} \in \mathbb{R}^n_+ \mid \boldsymbol{Ax} = \boldsymbol{b}\}$ and $S = P \cap \mathbb{Z}^n$.

The reformulation takes the form

$$\max\{\boldsymbol{cx}^0 + \boldsymbol{cQ\mu} \mid \boldsymbol{Q\mu} \geq -\boldsymbol{x}^0,\ \boldsymbol{\mu} \in \mathbb{Z}^{n-m}\}, \tag{1.2}$$

*Received by the editors October 3, 2019; accepted for publication (in revised form) May 31, 2021; published electronically October 25, 2021.
https://doi.org/10.1137/19M1291145
**Funding:** The research has been financed in part by The Netherlands Organisation for Scientific Research, NWO, grant 613.000.801, which we gratefully acknowledge.
†Delft Institute of Applied Mathematics, TU Delft, The Netherlands, and Centrum Wiskunde en Informatica, Amsterdam, The Netherlands (k.i.aardal@tudelft.nl).
‡CERC, Polytechnique Montréal, Montréal H3C 3A7, QC, Canada, and Jacobs Technion-Cornell Institute, Cornell Tech and Technion – IIT, New York, NY 10044, USA (andrea.lodi@cornell.edu).
§Work done while at IBM CPLEX Optimization, IBM, Italy. Current address: FICO Xpress Optimization, FICO, Italy (andreatramontani@fico.com).
¶Work done while at University of Cologne, Cologne, Germany (mail@fredvonheymann.eu).
‖CORE, LIDAM, UCLouvain, Ottignies-Louvain-la-Neuve, Belgium (laurence.wolsey@uclouvain.be).

where $\boldsymbol{Q}$ is an $n\times(n-m)$ integer matrix and $\boldsymbol{x}^0$ is a point satisfying $\boldsymbol{A}\boldsymbol{x}^0=\boldsymbol{b},\ \boldsymbol{x}^0\in\mathbb{Z}^n$. Here we let $\hat{P}=\{\boldsymbol{\mu}\in\mathbb{R}^{n-m}:\boldsymbol{Q}\boldsymbol{\mu}\geq-\boldsymbol{x}^0\}$ and $\hat{S}=\hat{P}\cap\mathbb{Z}^{n-m}$. The integer sets $S$ and $\hat{S}$ are related: $\boldsymbol{x}\in S$ if and only if there exists $\boldsymbol{\mu}\in\hat{S}$ with $\boldsymbol{x}=\boldsymbol{x}^0+\boldsymbol{Q}\boldsymbol{\mu}$, or in other words $S=\mathrm{proj}_x\{(\boldsymbol{x},\boldsymbol{\mu}):\boldsymbol{x}=\boldsymbol{x}^0+\boldsymbol{Q}\boldsymbol{\mu},\boldsymbol{\mu}\in\hat{S}\}$.

Our idea is based on the computational experience with branch and bound on $\hat{P}$ rather than on $P$. Branching in unit directions on $\hat{P}$ has proven to be computationally more effective for certain problem types; see, e.g., [4, 1]. Here, we generate Gomory mixed-integer (GMI) cuts [24] from $\hat{P}$, which are not necessarily tableau cuts for $P$ but still computationally easy to generate.

A first practical observation is that if one considers the reformulated problem (1.2), one can generate $(n-m)$ potentially different Chvátal–Gomory (CG) [12] or GMI cuts off an optimal linear program (LP) tableau. Here we will concentrate on GMI cuts (also viewed as split cuts [14]) that will be called (lattice) *$\ell$-cuts*. The study of sets $P,\hat{P},S$, $\hat{S}$ and the proposed $\ell$-cuts raises a series of questions both theoretical and computational. For example:

- What is the relationship between $P$ and $\hat{P}$?
- Given a point $\boldsymbol{\mu}\in\hat{S}$, what is the corresponding point $\boldsymbol{x}\in S$, and vice versa?
- How strong are the $\ell$-cuts when expressed in the $\boldsymbol{x}$-space?
- Are the corner polyhedra associated to a basis in the $\boldsymbol{x}$ and $\boldsymbol{\mu}$ spaces the same?
- What, if any, is the relationship between the GMI or split closures of $P$ and $\hat{P}$?

Computational questions that we investigate are:

- How effective are the $\ell$-cuts?
- Can the $\ell$-cuts associated to a basis tableau be easily generated in the $\boldsymbol{x}$-space?

Lattice reformulation of (mixed) integer optimization problems was introduced by Lenstra Jr. [31]. For an overview of results on lattice reformulations and integer programming we refer to [2]. For some articles, especially related to computational aspects, see [15, 32, 28, 5].

We now point to some related computational work on cutting planes. Bonami et al. [10] observed that one round of GMI inequalities generated from an optimal basic solution closed 24% of the integrality gap on average on 43 mixed integer MIPLIB 3.0 [8] instances. Cornuéjols et al. [17] suggested to multiply a row in the optimal LP tableau by an integer $k$ and then derive a GMI off of the resulting row. They called a cut generated in this way a $k$-cut. The standard tableau GMI inequality is a $k$-cut with $k=1$. One motivation behind this approach is to create a large fractional right-hand side of the resulting tableau row as this intuitively could lead to a stronger inequality. Whether the inequality actually is stronger of course depends on the left-hand side coefficients as well. Later Cornuéjols [16] suggested that one should look for deep split cuts that can be separated efficiently. This is also the viewpoint taken here.

An alternative, but very costly approach, is to generate all the inequalities from a given family, known as the closure. Balas and Saxena [7] performed a computational study of the split inequalities and concluded that the split closure closed 73% of the integrality gap, on average, on 41 mixed integer MIPLIB 3.0 instances, and 72%, on average, on 24 pure integer MIPLIB 3.0 instances. It is, however, NP-hard to optimize a linear function over the split closure [11], so achieving these results is computationally expensive. Of course, a vast literature has been devoted to computationally viable ways of approximating the split closure; see, e.g., Dash and Goycoolea [19] and Fischetti and Salvagnin [23].

Our research aims to investigate $\ell$-cuts, which can be separated in polynomial time, and their effectiveness in improving on the bounds obtained from easy-to-generate GMIs, such as tableau GMIs and $k$-cuts.

In section 2 we present the background material we need concerning inequalities and lattices. In section 3 we see that most of the theoretical questions have simple and perhaps surprising answers. In particular, even though the GMI/split cuts generated may be different, the GMI/split closures are the same. We give a description of our approach for generating violated inequalities in section 4 and present our computational results comparing different possible variants in section 5. Finally, some conclusions are drawn in section 6.

## 2. Background.

**2.1. GMI inequalities and split inequalities.** We define GMI inequalities and split inequalities, $k$-cuts, and closures. For a more general exposition we refer the reader to [13, 34].

Consider the single row mixed-integer set

$$X = \left\{ (\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{Z}_+^n \times \mathbb{R}_+^p \mid \sum_{j=1}^n a_j x_j + \sum_{j=1}^p g_j y_j = b \right\} \tag{2.1}$$

and suppose that $b \notin \mathbb{Z}$. Let

$b := \lfloor b \rfloor + f_0$ with $0 < f_0 < 1$,

$a_j := \lfloor a_j \rfloor + f_j$ with $0 \le f_j < 1$.

The *GMI inequality* [24] for $X$ is

$$\sum_{\{j: f_j \le f_0\}} \frac{f_j}{f_0} x_j + \sum_{\{j: f_j > f_0\}} \frac{1 - f_j}{1 - f_0} x_j + \sum_{\{j: g_j > 0\}} \frac{g_j}{f_0} y_j - \sum_{\{j: g_j < 0\}} \frac{g_j}{1 - f_0} y_j \ge 1\,. \tag{2.2}$$

If the row (2.1) is a row from a simplex tableau of a linear relaxation, the associated GMI inequality is referred to as a *tableau GMI inequality*.

Cornuéjols et al. [17] introduced *k-cuts*, which are cuts that are obtained by first multiplying (2.1) from an optimal tableau in which one of the $\boldsymbol{x}$-variables is basic by an integer $k$ and then deriving the GMI inequality. In this paper we introduce $\ell$-cuts, which are tableau GMI cuts derived from an optimal tableau of the LP-relaxation of (1.2). In sections 3 and 4 we explain how to generate these cuts in the space of the $\boldsymbol{x}$-variables.

Let $T$ be a polyhedron in $\mathbb{R}^{n+p}$. Next, we consider a mixed integer set $T \cap (\mathbb{Z}^n \times \mathbb{R}^p)$. For a given $(\boldsymbol{\pi}, \pi_0) \in \mathbb{Z}^{n+1}$ we define

$$\Pi_1 := T \cap \{(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{Z}^n \times \mathbb{R}^p \mid \boldsymbol{\pi x} \le \pi_0\},$$
$$\Pi_2 := T \cap \{(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{Z}^n \times \mathbb{R}^p \mid \boldsymbol{\pi x} \ge \pi_0 + 1\}\,.$$

An inequality $\boldsymbol{\alpha x} + \boldsymbol{\gamma y} \le \beta$ is called a *split inequality* [14] if there exists a $(\boldsymbol{\pi}, \pi_0) \in \mathbb{Z}^{n+1}$ such that $\boldsymbol{\alpha x} + \boldsymbol{\gamma y} \le \beta$ is valid for $\Pi_1 \cup \Pi_2$. The disjunction $\boldsymbol{\pi x} \le \pi_0 \vee \boldsymbol{\pi x} \ge \pi_0 + 1$ is called a *split disjunction*. The GMI inequality can be viewed as a split inequality for (2.1) with the split in which $\pi_j = \lfloor a_j \rfloor$ if $f_j \le f_0$, $\pi_j = \lceil a_j \rceil$ if $f_j > f_0$ and $\pi_0 = \lfloor b \rfloor$.

The *elementary closure*, or simply the *closure*, associated with a family $F$ of inequalities valid for $T \cap (\mathbb{Z}^n \times \mathbb{R}^p)$ is the convex set obtained as the intersection of all inequalities in $F$. It is known that the split closure and the GMI closure are equivalent [35] and that the separation problem for the split closure is NP-hard [11].

Observation 2.1. *If $X$ is replaced by*

$$\bar{X} = \left\{ (\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{Z}_+^n \times \mathbb{R}_+^p \mid \sum_{j=1}^{n} \bar{a}_j x_j + \sum_{j=1}^{p} g_j y_j = \bar{b} \right\},$$

*where $\bar{a}_j \equiv a_j \mod 1$ for $1 \leq j \leq n$ and $\bar{b} \equiv b \mod 1$, the GMI* (2.2) *for $X$ and the GMI for $\bar{X}$ are the same inequality.*

**2.2. Lattices and lattice reformulation.** Given $l \leq n$ linearly independent vectors $\boldsymbol{b}_1, \ldots \boldsymbol{b}_l \in \mathbb{R}^n$, the set $L(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_l) := \{\sum_{i=1}^{l} c_i \boldsymbol{b}_i,\ c_i \in \mathbb{Z}\}$ is called the *lattice* generated by $\boldsymbol{b}_1, \ldots \boldsymbol{b}_l$. The vectors $\boldsymbol{b}_1, \ldots \boldsymbol{b}_l$ are called a *lattice basis*, and we often represent them as a matrix $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots \boldsymbol{b}_l)$. Given a lattice $L$ generated by $\boldsymbol{B}$, the basis $\boldsymbol{B}'$ is an alternative basis for $L$ if and only if we can write $\boldsymbol{B}' = \boldsymbol{B}\boldsymbol{U}$, where $\boldsymbol{U}$ is an $l \times l$ unimodular matrix.

We now can explain the reformulation of the IP

$$\max\{\boldsymbol{c}\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \in \mathbb{Z}_+^n\}, \tag{1.1}$$

presented in section 1, due to Aardal et al. [3]. The set $\ker_{\mathbb{Z}}(\boldsymbol{A}) = \{\boldsymbol{x} \in \mathbb{Z}^n \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}\}$ is a lattice, called the *kernel lattice* of $\boldsymbol{A}$.

Suppose $\boldsymbol{x}$ is a feasible solution in (1.1). If $\boldsymbol{x}^0 \in \mathbb{Z}^n$ satisfies $\boldsymbol{A}\boldsymbol{x}^0 = \boldsymbol{b}$, it follows that $\boldsymbol{A}(\boldsymbol{x} - \boldsymbol{x}^0) = \boldsymbol{0}$ and thus, if $\boldsymbol{Q}$ is a lattice basis for $\ker_{\mathbb{Z}}(\boldsymbol{A})$, this is equivalent to $(\boldsymbol{x} - \boldsymbol{x}^0) = \boldsymbol{Q}\boldsymbol{\mu}$, where $\boldsymbol{\mu} \in \mathbb{Z}^{n-m}$. Now substituting $\boldsymbol{x} = \boldsymbol{x}^0 + \boldsymbol{Q}\boldsymbol{\mu}$ and using $\boldsymbol{x} \geq \boldsymbol{0}$ gives the reformulation

$$\max\{\boldsymbol{c}(\boldsymbol{x}^0 + \boldsymbol{Q}\boldsymbol{\mu}) \mid \boldsymbol{Q}\boldsymbol{\mu} \geq -\boldsymbol{x}^0, \boldsymbol{\mu} \in \mathbb{Z}^{n-m}\}. \tag{1.2}$$

Let $L$ be a lattice in a Euclidean vector space $E$. A subset $K \subseteq L$ is called a *pure sublattice* of $L$ if there exists a linear subspace $D$ of $E$ such that $K = D \cap L$.

A matrix $\boldsymbol{A} \in \mathbb{Z}^{m \times n}$ of full row rank is in *Hermite normal form* if it has the form $\text{HNF}(\boldsymbol{A}) = (\boldsymbol{H}, \boldsymbol{0}^{m \times (n-m)}) = \boldsymbol{A}\boldsymbol{U}$, where $\boldsymbol{H}$ is a lower triangular nonnegative $m \times m$ matrix in which the unique row maxima can be found along the diagonal, and $\boldsymbol{U}$ is an $n \times n$ unimodular matrix.

Observation 2.2. *A lattice $L$ generated by the basis $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_l)$ is a pure sublattice of the standard lattice $\mathbb{Z}^n$ if and only if* $\text{HNF}(\boldsymbol{B}^\mathsf{T}) = (\boldsymbol{I}, \boldsymbol{0})$.

Observation 2.3. *The lattice* $\ker_{\mathbb{Z}}(\boldsymbol{A})$ *is a pure sublattice of $\mathbb{Z}^n$.*

Theorem 2.4 (see Schrijver [37, Theorem 5.2]). *The Hermite normal form $(\boldsymbol{H}, \boldsymbol{0})$ of a rational matrix $\boldsymbol{A}$ of full row rank has size polynomially bounded by the size of $\boldsymbol{A}$. Moreover, there exists a unimodular matrix $\boldsymbol{U}$ with $\boldsymbol{A}\boldsymbol{U} = (\boldsymbol{H}, \boldsymbol{0})$ such that the size of $\boldsymbol{U}$ is polynomially bounded by the size of $\boldsymbol{A}$.*

Proposition 2.5 (see Schrijver [37, Corollary 5.3a]). *Given a rational matrix $\boldsymbol{A}$ of full row rank, a unimodular matrix $\boldsymbol{U}$ such that $\boldsymbol{A}\boldsymbol{U}$ is in Hermite normal form can be found in polynomial time.*

**3. Relations between solutions and polyhedra in $\boldsymbol{x}$- and $\boldsymbol{\mu}$-space.** Here we establish answers to the theoretical questions raised in the introduction.

**3.1. Expressing $\boldsymbol{\mu} \in \hat{S}$ as a function of $\boldsymbol{x} \in S$.** The lattice reformulation gives a way of expressing each feasible vector $\boldsymbol{x} \in S$ as a function of $\boldsymbol{\mu}$. A natural question is how to express a feasible vector $\boldsymbol{\mu} \in \hat{S}$ as a function of $\boldsymbol{x}$. In particular, this is our prime tool for generating general disjunctions for deriving split inequalities, as described in more detail in section 4. Notice that, if the full instance is reformulated, it would of course be possible to perform all computations using the reformulation only. However, in a more general context, one could have large and heterogeneous problems and reformulate only one part of the constraints and of the variables—those that correspond to a clean structure on which the reformulation is expected to be effective. In addition, it is often desirable to work in the $\boldsymbol{x}$-space as the meaning of these variables is clear to the user. Those considerations motivate us to derive a way to translate information from the $\boldsymbol{\mu}$-space back to the $\boldsymbol{x}$-space.

A consequence of $\ker_{\mathbb{Z}}(\boldsymbol{A})$ being a pure sublattice of $\mathbb{Z}^n$, and of Theorem 2.4 and Proposition 2.5, is that we can find, in polynomial time, a unimodular matrix $\boldsymbol{U}$ such that

$$\boldsymbol{U}^{\mathsf{T}}\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{I} \\ \boldsymbol{0} \end{pmatrix}. \tag{3.1}$$

Let $\boldsymbol{W}$ be the matrix consisting of the first $n-m$ rows of $\boldsymbol{U}^{\mathsf{T}}$ as in (3.1). Since $\boldsymbol{W}$ is a submatrix of $\boldsymbol{U}^{\mathsf{T}}$ it follows that all elements of $\boldsymbol{W}$ are integral. It is also clear that

$$\boldsymbol{W}\boldsymbol{Q} = \boldsymbol{I}\,. \tag{3.2}$$

This was also observed by Mehrotra and Li [33]. Note that $\boldsymbol{W}$ in general is not unique: given a matrix $\boldsymbol{W}$, we can form a matrix $\boldsymbol{W}' = \boldsymbol{W} + \boldsymbol{C}$, where $\boldsymbol{C}$ is an integer $(n-m)\times n$ matrix consisting of rows obtained by taking an integer linear combination of rows of $\boldsymbol{A}$. The matrix $\boldsymbol{W}$ permits us to translate an expression in $\boldsymbol{\mu}$-variables back to an expression in $\boldsymbol{x}$-variables. Specifically we have $\boldsymbol{W}\boldsymbol{x} = \boldsymbol{W}\boldsymbol{x}^0 + \boldsymbol{W}\boldsymbol{Q}\boldsymbol{\mu}$, and thus

$$\boldsymbol{\mu} = \boldsymbol{W}\boldsymbol{x} - \boldsymbol{W}\boldsymbol{x}^0\,.$$

**3.2. Relations between bases and polyhedra in the $\boldsymbol{x}$- and $\boldsymbol{\mu}$-spaces.** Let $\boldsymbol{B}$ be an $m\times m$ nonsingular submatrix of $\boldsymbol{A}$. Such a basis exists as $rank(\boldsymbol{A}) = m$. Given basis $\boldsymbol{B}$, we examine the corresponding partitions of $\boldsymbol{A}, \boldsymbol{x}, \boldsymbol{Q}$, and $\boldsymbol{W}$.

PROPOSITION 3.1. *Given $\boldsymbol{A}, \boldsymbol{Q}, \boldsymbol{W}$ as described above and a basis $\boldsymbol{B}$, write $\boldsymbol{A} = (\boldsymbol{B}, \boldsymbol{N})$, $\boldsymbol{Q}^{\mathsf{T}} = (\boldsymbol{Q}_B, \boldsymbol{Q}_N)$, $\boldsymbol{x} = (\boldsymbol{x}_B, \boldsymbol{x}_N)$, $\boldsymbol{W} = (\boldsymbol{W}_B, \boldsymbol{W}_N)$, and $\boldsymbol{B}\boldsymbol{x}_B^0 + \boldsymbol{N}\boldsymbol{x}_N^0 = \boldsymbol{b}$. The following hold:*

(i) $\boldsymbol{Q}_B = -\boldsymbol{B}^{-1}\boldsymbol{N}\boldsymbol{Q}_N$,

(ii) *$\boldsymbol{Q}_N$ is nonsingular, and $\boldsymbol{Q}_N^{-1} = \boldsymbol{W}_N - \boldsymbol{W}_B\boldsymbol{B}^{-1}\boldsymbol{N}$.*

*Proof.* (i) As $\boldsymbol{A}\boldsymbol{Q} = \boldsymbol{0}$, $\boldsymbol{B}\boldsymbol{Q}_B + \boldsymbol{N}\boldsymbol{Q}_N = \boldsymbol{0}$, and, as $\boldsymbol{B}^{-1}$ exists, $\boldsymbol{Q}_B = -\boldsymbol{B}^{-1}\boldsymbol{N}\boldsymbol{Q}_N$.

(ii) As $\boldsymbol{W}\boldsymbol{Q} = \boldsymbol{I}$, $\boldsymbol{W}_B\boldsymbol{Q}_B + \boldsymbol{W}_N\boldsymbol{Q}_N = \boldsymbol{I}$, and using (i), one has $(-\boldsymbol{W}_B\boldsymbol{B}^{-1}\boldsymbol{N} + \boldsymbol{W}_N)\boldsymbol{Q}_N = \boldsymbol{I}$. It follows as $\boldsymbol{Q}_N$ and $\boldsymbol{W}_N - \boldsymbol{W}_B\boldsymbol{B}^{-1}\boldsymbol{N}$ are both $(n-m)\times(n-m)$ matrices that $\boldsymbol{Q}_N$ is nonsingular and thus $\boldsymbol{Q}_N^{-1} = \boldsymbol{W}_N - \boldsymbol{W}_B\boldsymbol{B}^{-1}\boldsymbol{N}$. ∎

Now we show that not only vectors in $S$ and $\hat{S}$ correspond one to one, but that there is also a one-to-one correspondence between vectors in $P$ and $\hat{P}$.

PROPOSITION 3.2. *Given $\boldsymbol{A} \in \mathbb{Z}^{m\times n}$ and $\boldsymbol{b} \in \mathbb{Z}^m$, define $P = \{\boldsymbol{x} \in \mathbb{R}_+^n \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}\}$. Define $\hat{P} = \{\boldsymbol{\mu} \in \mathbb{R}^{n-m} \mid \boldsymbol{Q}\boldsymbol{\mu} \geq -\boldsymbol{x}^0\}$ for $\boldsymbol{Q}$ and $\boldsymbol{x}^0$ as given above. The map $f(\boldsymbol{\mu}) = \boldsymbol{Q}\boldsymbol{\mu} + \boldsymbol{x}^0$ is a bijective map from $\hat{P}$ to $P$.*

*Proof.* Take $\bar{\boldsymbol{\mu}} \in \hat{P}$, and let $\bar{\boldsymbol{x}} = \boldsymbol{Q}\bar{\boldsymbol{\mu}} + \boldsymbol{x}^0$. The vector $\bar{\boldsymbol{x}}$ is nonnegative since $\boldsymbol{Q}\bar{\boldsymbol{\mu}} \geq -\boldsymbol{x}^0$. Moreover, $\boldsymbol{A}\bar{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{Q}\bar{\boldsymbol{\mu}} + \boldsymbol{A}\boldsymbol{x}^0 = \boldsymbol{A}\boldsymbol{x}^0 = \boldsymbol{b}$, where the second equality holds since $\boldsymbol{Q}$ is a basis for $\ker_{\mathbb{Z}}(\boldsymbol{A})$.

Take $\bar{\boldsymbol{x}} \in P$. Since $\boldsymbol{Q}$ spans the Euclidean vector space $\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}\}$, we can write $\bar{\boldsymbol{x}}$ as $\bar{\boldsymbol{x}} = \boldsymbol{Q}\bar{\boldsymbol{\mu}} + \boldsymbol{x}^0$ for some $\bar{\boldsymbol{\mu}} \in \mathbb{R}^{n-m}$. $\bar{\boldsymbol{x}} \in P$ implies $\bar{\boldsymbol{x}} \geq \boldsymbol{0}$, and hence $\boldsymbol{Q}\bar{\boldsymbol{\mu}} + \boldsymbol{x}^0 \geq \boldsymbol{0}$, so $\bar{\boldsymbol{\mu}} \in \hat{P}$. Thus the function $f$ is injective.

Finally consider $\boldsymbol{\mu}^1, \boldsymbol{\mu}^2 \in \hat{P}$ with images $\boldsymbol{x}^i = \boldsymbol{Q}\mu^i + \boldsymbol{x}^0$ for $i = 1, 2$. If $\boldsymbol{x}^1 = \boldsymbol{x}^2$, then $\boldsymbol{Q}(\boldsymbol{\mu}^1 - \boldsymbol{\mu}^2) = \boldsymbol{0}$ implying that $\boldsymbol{Q}_N(\boldsymbol{\mu}^1 - \boldsymbol{\mu}^2) = \boldsymbol{0}$, where $\boldsymbol{Q}_N$ is as in Proposition 3.1. But from the Proposition, $\boldsymbol{Q}_N$ is nonsingular, and thus $\boldsymbol{\mu}^1 = \boldsymbol{\mu}^2$ and the function $f$ is also surjective. ∎

Now we consider the representation of the basis in the $\boldsymbol{x}$- and $(\boldsymbol{x}, \boldsymbol{\mu})$-spaces. A basic solution in the $\boldsymbol{x}$-space is written as

$$\boldsymbol{x}_B + \boldsymbol{B}^{-1}\boldsymbol{N}\boldsymbol{x}_N = \boldsymbol{B}^{-1}\boldsymbol{b}, \quad \boldsymbol{x}_B, \boldsymbol{x}_N \geq \boldsymbol{0}\,.$$

Consider the polyhedron $\hat{P}$ as defined in Proposition 3.2. If we write the constraints defining $\hat{P}$ in equality form we notice, from the definition of the reformulation, that the $\boldsymbol{x}$-variables are precisely the slack variables, i.e., $\boldsymbol{Q}\boldsymbol{\mu} - \boldsymbol{x} = -\boldsymbol{x}^0$, or equivalently,

$$\boldsymbol{x} - \boldsymbol{Q}\boldsymbol{\mu} = \boldsymbol{x}^0\,,$$

which we can write, in the $(\boldsymbol{x}, \boldsymbol{\mu})$-space as

$$\begin{pmatrix} \boldsymbol{x}_B \\ \boldsymbol{x}_N \end{pmatrix} - \begin{pmatrix} \boldsymbol{Q}_B \\ \boldsymbol{Q}_N \end{pmatrix} \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{x}^0_B \\ \boldsymbol{x}^0_N \end{pmatrix}. \tag{3.3}$$

OBSERVATION 3.3. *In a basic feasible solution of* (3.3)*, all the* $\boldsymbol{\mu}$*-variables are basic as they are free variables. In addition,* $m$ *of the slack variables, i.e.,* $m$ *of the original* $\boldsymbol{x}$*-variables, are basic.*

Hence, the basic variables are $(\boldsymbol{x}_B, \boldsymbol{\mu})$. Multiplying the last $n - m$ rows of (3.3) by $-\boldsymbol{Q}_N^{-1}$ yields

$$-\boldsymbol{Q}_N^{-1}\boldsymbol{x}_N + \boldsymbol{I}\boldsymbol{\mu} = -\boldsymbol{Q}_N^{-1}\boldsymbol{x}^0_N\,, \text{ or equivalently } \boldsymbol{\mu} = \boldsymbol{Q}_N^{-1}\boldsymbol{x}_N - \boldsymbol{Q}_N^{-1}\boldsymbol{x}^0_N\,.$$

Substituting for $\boldsymbol{\mu}$ in the first $m$ rows of (3.3) gives

$$\boldsymbol{x}_B - \boldsymbol{Q}_B\boldsymbol{Q}_N^{-1}\boldsymbol{x}_N = \boldsymbol{x}^0_B - \boldsymbol{Q}_B\boldsymbol{Q}_N^{-1}\boldsymbol{x}^0_N\,,$$

and we obtain an expression for a basic solution:

$$\begin{pmatrix} \boldsymbol{x}_B \\ \boldsymbol{\mu} \end{pmatrix} - \begin{pmatrix} \boldsymbol{Q}_B\boldsymbol{Q}_N^{-1} \\ \boldsymbol{Q}_N^{-1} \end{pmatrix} \boldsymbol{x}_N = \begin{pmatrix} \boldsymbol{x}^0_B - \boldsymbol{Q}_B\boldsymbol{Q}_N^{-1}\boldsymbol{x}^0_N \\ -\boldsymbol{Q}_N^{-1}\boldsymbol{x}^0_N \end{pmatrix}. \tag{3.4}$$

Now, using Proposition 3.1, the basic solution (3.4) can be rewritten as

$$\begin{aligned} \begin{pmatrix} \boldsymbol{x}_B \\ \boldsymbol{\mu} \end{pmatrix} + \begin{pmatrix} \boldsymbol{B}^{-1}\boldsymbol{N} \\ -(\boldsymbol{W}_N - \boldsymbol{W}_B\boldsymbol{B}^{-1}\boldsymbol{N}) \end{pmatrix} \boldsymbol{x}_N &= \begin{pmatrix} \boldsymbol{x}^0_B + \boldsymbol{B}^{-1}\boldsymbol{N}\boldsymbol{x}^0_N \\ -(\boldsymbol{W}_N - \boldsymbol{W}_B\boldsymbol{B}^{-1}\boldsymbol{N})\boldsymbol{x}^0_N \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{B}^{-1}\boldsymbol{b} \\ -(\boldsymbol{W}_N - \boldsymbol{W}_B\boldsymbol{B}^{-1}\boldsymbol{N})\boldsymbol{x}^0_N \end{pmatrix}. \qquad (3.5) \end{aligned}$$

From (3.4) we see that, given a basis, the $\boldsymbol{\mu}$-variables can be expressed solely as a function of $\boldsymbol{Q}$.

We now illustrate the different basis representations in an example.

*Example* 3.4. Consider an instance with $m = 2,\ n = 5$:

$$(\boldsymbol{A}\,|\,\boldsymbol{b}) = \left( \begin{array}{ccccc|c} 0 & 5 & 3 & 1 & 7 & 9 \\ 6 & 3 & 0 & 11 & 2 & 14 \end{array} \right).$$

To obtain a reformulation, one can take

$$\boldsymbol{Q} = \begin{pmatrix} 1 & -3 & -3 \\ 3 & 3 & 0 \\ 0 & -3 & 4 \\ -1 & 1 & 2 \\ -2 & -1 & -2 \end{pmatrix}, \quad \boldsymbol{x}^0 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$

A matrix $\boldsymbol{W}$ corresponding to $\boldsymbol{Q}$ is

$$\boldsymbol{W} = \begin{pmatrix} -2 & -1 & 0 & -4 & -1 \\ -2 & 1 & 1 & -3 & 2 \\ -3 & 0 & 1 & -5 & 1 \end{pmatrix}.$$

For the feasible basis $\boldsymbol{x}_B = (x_1, x_2)$, the corresponding $\boldsymbol{x}$-tableau is

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \frac{1}{30} \begin{pmatrix} -9 & 52 & -11 \\ 18 & 6 & 42 \end{pmatrix} \begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix} = \frac{1}{30} \begin{pmatrix} 43 \\ 54 \end{pmatrix}.$$

Now setting $\boldsymbol{\mu} = \boldsymbol{W}\boldsymbol{x} - \boldsymbol{W}\boldsymbol{x}^0$ and eliminating the basic variables $\boldsymbol{x}_B$ by substitution, the corresponding $(\boldsymbol{x}, \boldsymbol{\mu})$-tableau consists of the above $\boldsymbol{x}$-tableau plus

$$\begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix} - \frac{1}{30} \begin{pmatrix} 0 & -10 & -10 \\ -6 & 8 & -4 \\ 3 & 6 & -3 \end{pmatrix} \begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix} = \frac{1}{30} \begin{pmatrix} 10 \\ -2 \\ -9 \end{pmatrix}.$$

From the $\mu_3$ row, one has $f_3 = \frac{27}{30}, f_4 = \frac{24}{30}, f_5 = \frac{3}{30}$, and $f_0 = \frac{21}{30}$ giving the $\ell$-cut:

$$\frac{1}{3}x_3 + \frac{2}{3}x_4 + \frac{1}{7}x_5 \geq 1.$$

We now turn our attention to the group problem associated with the formulations and the related corner polyhedra [25]. Let $\boldsymbol{A} = (\boldsymbol{B},\ \boldsymbol{N})$, where $\boldsymbol{B}$ corresponds to the basic variables in an optimal solution to the LP-relaxation of (1.1). The following integer optimization problem is a relaxation of (1.1) obtained by dropping the nonnegativity constraints on the basic variables $\boldsymbol{x}_B$:

$$\max\{\boldsymbol{c}\boldsymbol{x} \mid (\boldsymbol{B}\ \boldsymbol{N}) \begin{pmatrix} \boldsymbol{x}_B \\ \boldsymbol{x}_N \end{pmatrix} = \boldsymbol{b},\ \boldsymbol{x}_N \geq \boldsymbol{0},\ \boldsymbol{x}_B, \boldsymbol{x}_N \text{ integral}\}. \tag{3.6}$$

Using the relation $\boldsymbol{B}\boldsymbol{x}_B + \boldsymbol{N}\boldsymbol{x}_N = \boldsymbol{b}$ and the integrality of $\boldsymbol{x}_B$ gives the equivalent formulation of (3.6) as

$$\max\{\boldsymbol{c}_B\boldsymbol{B}^{-1}\boldsymbol{b} + (\boldsymbol{c}_N - \boldsymbol{c}_B\boldsymbol{B}^{-1}\boldsymbol{N})\boldsymbol{x}_N \mid \boldsymbol{B}^{-1}\boldsymbol{N}\boldsymbol{x}_N \equiv \boldsymbol{B}^{-1}\boldsymbol{b} \mod 1,\ \boldsymbol{x}_N \in \mathbb{Z}_+^{n-m}\}. \tag{3.7}$$

Problem (3.7) is referred to as the *group problem* [25].

We will now prove that the feasible sets of the group problem are the same whether we view them in the original $\boldsymbol{x}$-space or in the reformulated space.

THEOREM 3.5. *The groups*

$$G = \{\boldsymbol{x}_N \in \mathbb{Z}_+^{n-m} \mid \boldsymbol{B}^{-1}\boldsymbol{N}\boldsymbol{x}_N \equiv \boldsymbol{B}^{-1}\boldsymbol{b} \mod 1\}$$

*and*

$$\hat{G} = \{\boldsymbol{x}_N \in \mathbb{Z}_+^{n-m} \mid -(\boldsymbol{W}_N - \boldsymbol{W}_B\boldsymbol{B}^{-1}\boldsymbol{N})\boldsymbol{x}_N \equiv -(\boldsymbol{W}_N - \boldsymbol{W}_B\boldsymbol{B}^{-1}\boldsymbol{N})\boldsymbol{x}_N^0 \mod 1\}$$

*are the same.*

*Proof.* As $\boldsymbol{W}_N\boldsymbol{x}_N, \boldsymbol{W}_N\boldsymbol{x}_N^0$ are integer,

$$\hat{G} = \{\boldsymbol{x}_N \in \mathbb{Z}_+^{n-m} \mid \boldsymbol{W}_B\boldsymbol{B}^{-1}\boldsymbol{N}\boldsymbol{x}_N \equiv \boldsymbol{W}_B\boldsymbol{B}^{-1}\boldsymbol{N}\boldsymbol{x}_N^0 \mod 1\}. \tag{3.8}$$

Now as $\boldsymbol{W}_B$ is an integral matrix, it follows that $G \subseteq \hat{G}$.

Conversely, take $\hat{G}$ in the form (see (3.4)):

$$\hat{G} = \{\boldsymbol{x}_N \in \mathbb{Z}_+^{n-m} \mid \boldsymbol{Q}_N^{-1}\boldsymbol{x}_N \equiv \boldsymbol{Q}_N^{-1}\boldsymbol{x}_N^0 \mod 1\}.$$

Suppose $\boldsymbol{x}_N \in \hat{G}$. As $\boldsymbol{Q}_B$ is an integer matrix, $\boldsymbol{x}_N$ lies in

$$\{\boldsymbol{x}_N \in \mathbb{Z}_+^{n-m} \mid \boldsymbol{Q}_B\boldsymbol{Q}_N^{-1}\boldsymbol{x}_N \equiv \boldsymbol{Q}_B\boldsymbol{Q}_N^{-1}\boldsymbol{x}_N^0 \mod 1\},$$

which, as $\boldsymbol{Q}_B\boldsymbol{Q}_N^{-1} = -\boldsymbol{B}^{-1}\boldsymbol{N}$, is precisely $G$. ◻

As the order of the groups is given by the determinant, it follows that $|\det(\boldsymbol{B})| = |\det(\boldsymbol{Q}_N)|$, and as the corner polyhedron is the convex hull of the solutions to the group problem, it follows immediately that the corner polyhedra are the same.

Based on Observation 2.1, we see that the $\ell$-cuts generated from the second set of equations of (3.4), the second set of equations of (3.5) or from (3.8) are the same.

OBSERVATION 3.6. *Taking* $\boldsymbol{\mu} = \boldsymbol{W}(\boldsymbol{x} - \boldsymbol{x}^0)$ *or* $\boldsymbol{\mu}' = \boldsymbol{W}_B(\boldsymbol{x}_B - \boldsymbol{x}_B^0)$ *leads to the same $\ell$-cuts because* $\boldsymbol{W}_N\boldsymbol{x}_N \equiv 0 \mod 1$ *and* $\boldsymbol{W}_N\boldsymbol{x}_N^0 \equiv 0 \mod 1$. *Therefore a simple way to obtain the $\ell$-cuts is to left multiply the $\boldsymbol{x}$-tableau by $\boldsymbol{W}_B$. It follows that $\boldsymbol{W}_B$ is an $m$-dimensional generalization of the $k$ in $k$-cuts. In particular, if $m = 1$, the $(n-1)$ integer entries of $\boldsymbol{W}_B$ provide us with $(n-1)$ $k$-cuts. On the other hand, if $m > 1$, the $\ell$-cuts can be viewed as multirow tableau cuts; see, e.g.,* [18].

Now we consider closures. Let $P_S$ ($P_{CG}$) be the split (CG) closure with respect to $P$. Analogous notation is used for $\hat{P}$. We show that the split closures associated with $P$ and $\hat{P}$ are equivalent.

THEOREM 3.7. $P_{\mathrm{S}} = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} = \boldsymbol{x}^0 + \boldsymbol{Q}\boldsymbol{\mu},\ \boldsymbol{\mu} \in \hat{P}_{\mathrm{S}}\}$.

*Proof.* We use the definition of split cuts from [14]. Let $(\boldsymbol{\pi}, \pi_0) \in \mathbb{Z}^{n+1}$, and let

$$\boldsymbol{\alpha x} - q(\boldsymbol{\pi x} - \pi_0) \le \alpha_0,$$
$$\boldsymbol{\alpha x} + r(\boldsymbol{\pi x} - \pi_0 - 1) \le \alpha_0$$

be valid inequalities for $P$ with $q, r \ge 0$. Then, $\boldsymbol{\alpha x} \le \alpha_0$ is valid for $(P \cap \{\boldsymbol{\pi x} \le \pi_0\}) \cup (P \cap \{\boldsymbol{\pi x} \ge \pi_0 + 1\})$. The inequality $\boldsymbol{\alpha x} \le \alpha_0$ is called a split cut.

Substitute $\boldsymbol{x}$ for $\boldsymbol{Q\mu} + \boldsymbol{x}^0$. Let

$$\hat{\boldsymbol{\pi}} = \boldsymbol{\pi Q},$$
$$\hat{\pi}_0 = \pi_0 - \boldsymbol{\pi x}^0,$$
$$\hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha Q},$$
$$\hat{\alpha_0} = \alpha_0 - \boldsymbol{\alpha x}^0.$$

Notice that $(\hat{\boldsymbol{\pi}}, \hat{\pi}_0) \in \mathbb{Z}^{n-m+1}$ as $\boldsymbol{Q}$ and $\boldsymbol{x}^0$ are integer. We obtain

$$\hat{\boldsymbol{\alpha}}\boldsymbol{\mu} - q(\hat{\boldsymbol{\pi}}\boldsymbol{\mu} - \hat{\pi}_0) \leq \hat{\alpha_0}, \tag{3.9}$$

$$\hat{\boldsymbol{\alpha}}\boldsymbol{\mu} + r(\hat{\boldsymbol{\pi}}\boldsymbol{\mu} - \hat{\pi}_0 - 1) \leq \hat{\alpha_0}\,. \tag{3.10}$$

If inequalities (3.9) and (3.10) are valid for $\hat{P}$, then $\hat{\boldsymbol{\alpha}}\boldsymbol{\mu} \leq \hat{\alpha_0}$ is valid for $(\hat{P} \cap \{\hat{\boldsymbol{\pi}}\boldsymbol{\mu} \leq \hat{\pi}_0\}) \cup (\hat{P} \cap \{\hat{\boldsymbol{\pi}}\boldsymbol{\mu} \geq \hat{\pi}_0 + 1\})$.

Going from a split cut for $\hat{P}$ to a split cut for $P$ is similar by using $\boldsymbol{\mu} = \boldsymbol{W}(\boldsymbol{x} - \boldsymbol{x}^0)$ and using that $\boldsymbol{W}$ and $\boldsymbol{x}^0$ are integer. □

Our result also follows as a special case of Theorem 1 in Dash et al. [20], which was derived independently.

*A Chvátal–Gomory inequality* is a split inequality where one of the sets $(P \cap \{\boldsymbol{\pi x} \leq \pi_0\})$ or $(P \cap \{\boldsymbol{\pi x} \geq \pi_0 + 1\})$ is empty. The following result can be proved using the same method as in the proof of Theorem 3.7 together with Proposition 3.2.

Proposition 3.8. $P_{\mathrm{CG}} = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} = \boldsymbol{x}^0 + \boldsymbol{Q}\boldsymbol{\mu},\ \boldsymbol{\mu} \in \hat{P}_{\mathrm{CG}}\}$.

## 4. Separating cuts from lattice reformulations.

In subsection 4.1 we give a high-level description of our approach. In subsection 4.2 we describe three different reduction methods to derive the basis $\boldsymbol{Q}$ in the reformulation (1.2). In our computations we test how the quality of the reduction influences the effectiveness of the cuts generated. We also describe how to derive the matrix $\boldsymbol{W}$ in (3.2).

### 4.1. High-level description of our approach.

As discussed in section 3, $\ell$-cuts are tableau GMI cuts derived from an optimal tableau in the space of the $\boldsymbol{\mu}$-variables. However, they can be generated by working directly in the space of the $\boldsymbol{x}$-variables. The approach for separating $\ell$-cuts in the space of the $\boldsymbol{x}$-variables is as follows.

**Initialization:** Generate a reduced basis $\boldsymbol{Q}$ for $\ker_{\mathbb{Z}}(\boldsymbol{A})$ as in (1.2) and a corresponding matrix $\boldsymbol{W}$ (3.2) as shown in section 4.2.

**Iteration $t$:** After the addition of $t$ rounds of $\ell$-cuts,

1. Solve the resulting linear program and take the rows corresponding to the $\boldsymbol{x}$-variables in the basis. The resulting set of equations is of the form:

$$\begin{gathered}\boldsymbol{x}_B + \boldsymbol{N}\boldsymbol{x}_N + \boldsymbol{S}_N\boldsymbol{s}_N = \bar{\boldsymbol{x}}_B,\\ \boldsymbol{x}_B \in \mathbb{Z}_+^{|B|}, \boldsymbol{x}_N \in \mathbb{Z}_+^{|N|}, \boldsymbol{s}_N \geq \boldsymbol{0},\end{gathered} \tag{4.1}$$

   where $\boldsymbol{x}_N$ are the nonbasic $\boldsymbol{x}$-variables, $\boldsymbol{s}_N$ are nonbasic slack variables from previously added cuts, and $\boldsymbol{N}$ and $\boldsymbol{S}$ are the associated matrices in this part of the optimal tableau.
2. For every row $\boldsymbol{w}_i$ of $\boldsymbol{W}_B$ such that $\boldsymbol{w}_i\bar{\boldsymbol{x}}_B \notin \mathbb{Z}$, left multiply equation (4.1) by $\boldsymbol{w}_i$ to construct the "aggregated" tableau row

$$\boldsymbol{w}_i\boldsymbol{x}_B + \boldsymbol{w}_i\boldsymbol{N}\boldsymbol{x}_N + \boldsymbol{w}_i\boldsymbol{S}_N\boldsymbol{s}_N = \boldsymbol{w}_i\bar{\boldsymbol{x}}_B, \tag{4.2}$$

   generate the GMI cut from (4.2) (see (3.8)), and project out the slack variables $\boldsymbol{s}_N$ to get the cut in the space of the structural $\boldsymbol{x}$-variables only.
3. Add a selection of the separated cuts to the current LP.

In our implementation we used CPLEX 12.7.0 [26] as the LP solver. In order to avoid numerical issues and prevent separating invalid or numerically unstable cutting planes, we adopted several tolerances and safeguards. First, cutting planes are generated only from rows (4.2) where $\boldsymbol{w}_i\bar{\boldsymbol{x}}_B \notin \mathbb{Z}$ is fractional enough; i.e., $\boldsymbol{w}_i\bar{\boldsymbol{x}}_B - \lfloor \boldsymbol{w}_i\bar{\boldsymbol{x}}_B \rfloor \in\ ]\delta, 1-\delta[$ with $\delta = 10^{-3}$. Second, to limit numerical errors in the calculation of the aggregated row (4.2), the tableau rows (4.1) are not read from the final LP matrix factorization of CPLEX but are recalculated from scratch by aggregating the original rows with the optimal tableau multipliers given by the inverse of the basis matrix $\boldsymbol{B}^{-1}$. For cancelation of zero coefficients in (4.2), we use a tolerance $\epsilon = 10^{-10}$ and, as a further safeguard, we skip the row if the entries of the basic variables are different from the entries of $\boldsymbol{w}_i$, using the same tolerance $\epsilon = 10^{-10}$. Finally, we discard the GMI cuts separated from (4.2) if they have dynamism (i.e., ratio between the largest and the smallest absolute value of the nonzero coefficients) greater than $D = 10^8$, as they are deemed to be numerically unstable. We remark that all of these safeguards are quite customary, and they are applied to all different types of cuts that we discuss in section 5 (i.e., standard GMIs, $k$-cuts, and $\ell$-cuts).

**4.2. How to generate the matrices $\boldsymbol{Q}$ and $\boldsymbol{W}$.** The reformulation (1.2) is valid for any basis $\boldsymbol{Q}$ of the lattice $\ker_{\mathbb{Z}}(\boldsymbol{A})$. We will, however, be interested in a basis that is *reduced*. To test how the quality of the reduction plays a role in computations, we consider three different reductions.

**4.2.1. Lenstra–Lenstra–Lovász reductions.** Given linearly independent vectors $\boldsymbol{b}_1, \dots, \boldsymbol{b}_l \in \mathbb{R}^n$, the corresponding Gram–Schmidt orthogonalized vectors are

$$\begin{aligned} \boldsymbol{b}_1^* &= \boldsymbol{b}_1\,, \\ \boldsymbol{b}_j^* &= \boldsymbol{b}_j - \sum_{k=1}^{j-1} \mu_{jk}\boldsymbol{b}_k^*, \quad 2 \le j \le l\,, \quad \text{where} \\ \mu_{jk} &= \frac{\boldsymbol{b}_j^\top \boldsymbol{b}_k^*}{||\boldsymbol{b}_k^*||^2}, \quad 1 \le k < j \le l \end{aligned}$$

Definition 4.1 (Lenstra, Lenstra, Lovász [30]). *A basis $\boldsymbol{b}_1, \boldsymbol{b}_2, \dots, \boldsymbol{b}_l$ is called* LLL-reduced *if*

$$|\mu_{jk}| \le \frac{1}{2} \quad \textit{for } 1 \le k < j \le l\,, \tag{4.3}$$

$$||\boldsymbol{b}_j^* + \mu_{j,j-1}\boldsymbol{b}_{j-1}^*||^2 \ge y \cdot ||\boldsymbol{b}_{j-1}^*||^2 \quad \textit{for } 1 < j \le l \tag{4.4}$$

*for* $\frac{1}{4} < y < 1$.

Many quality guarantees can be given for a reduced basis. Well-known guarantees are that the first reduced basis vector is an approximation of the shortest nonzero vector in the lattice and that all reduced basis vectors are approximations of the successive minima of the lattice. We refer to [30] for details. A reduced basis can be computed in polynomial time, and the larger the parameter $y$ in (4.4), the better the quality guarantees become.

**4.2.2. Korkine–Zolotarev reduction.** A basis $\boldsymbol{b}_1, \dots \boldsymbol{b}_l$ of the lattice $L$ is reduced in the sense of Korkine and Zolotarev (KZ-reduced) [27] if it satisfies the following conditions:

1. $\boldsymbol{b}_1$ is a shortest nonzero vector of $L$ in the Euclidean norm,
2. $|\mu_{i1}| \le \frac{1}{2}$ for $2 \le i \le l$,
3. if $L^{(l-1)}$ denotes the orthogonal projection of $L$ on the orthogonal complement $(\mathbb{R}\boldsymbol{b}_1)^\perp$ of $\mathbb{R}\boldsymbol{b}_1$, then the projections $\boldsymbol{b}_i - \mu_{i1}\boldsymbol{b}_1$ of $\boldsymbol{b}_2, \dots, \boldsymbol{b}_l$ yield a KZ-reduced basis $\boldsymbol{b}_2 - \mu_{21}\boldsymbol{b}_1, \dots, \boldsymbol{b}_l - \mu_{l1}\boldsymbol{b}_1$ of $L^{(l-1)}$.

So, the first basis vector in a KZ-reduced basis is a shortest nonzero lattice vector. Several other bounds on the quality of such a basis, along with a nonrecursive definition of a KZ-reduced basis, can be found in [29]. Since a shortest lattice vector is computed, determining a KZ-reduced basis is computationally much more costly than determining an LLL-reduced basis.

In our computational study we test the following reduction methods.

**LLL-low:** LLL reduction with $y = 26/100$ to test a low-quality reduction.

**LLL:** LLL reduction with $y = 99/100$ to test a high-quality basis that is reasonably fast to compute.

**KZ:** KZ reduction to test in some sense an "optimally" reduced basis.

All the reductions are computed using the NTL library [38]. A possibility would have been to also include block KZ (BKZ) reduction [36] in our study. BKZ reduction is a hierarchy of reductions going from LLL-reduction to KZ reduction depending on the chosen block size. We chose not to include BKZ reduction as the computations indicated that the LLL reduction with $y = 99/100$ yields cutting plane results that are comparable with KZ reduction; see section 5.1.

#### 4.2.3. Computing the matrix $\boldsymbol{W}$.

As mentioned before, the matrix $\boldsymbol{W}$ is not unique. Let $\boldsymbol{e}_i$ be the $i$th column of the $(n-m)$-dimensional identity matrix. The matrix $\boldsymbol{W}$ can be calculated by computing the Hermite normal form as stated in Proposition 2.5. However, any method for finding a feasible solution to the $n-m$ systems of integer equations

$$\boldsymbol{Q}^{\mathsf{T}}\boldsymbol{w}_i = \boldsymbol{e}_i,\ \boldsymbol{w}_i \in \mathbb{Z}^n\ i = 1, \dots, n-m\,, \tag{4.5}$$

can be used. A valid matrix $\boldsymbol{W}$ is then obtained by taking the $n-m$ vectors $\boldsymbol{w}_i$ as its rows. In our computational study we again use the lattice reformulation technique described in [3] to derive the vectors $\boldsymbol{w}_i$, as this technique yields vectors $\boldsymbol{w}_i$ in which the absolute value of the elements is relatively small. For each of the $\boldsymbol{Q}$-matrices generated according to the three reductions given above, we generated an associated matrix $\boldsymbol{W}$, and the computations for (4.5) are all done using LLL reduction with $y = 99/100$.

From the perspective of theoretical runtime, the dominant part of our cutting plane algorithm is the generation of the matrix $\boldsymbol{Q}$ that is done in the initialization. Instead of generating $\boldsymbol{W}$ as described above, we could use expression (3.4), which only involves inverting $\boldsymbol{Q}_N$. The runtime of LLL reduction to generate $\boldsymbol{Q}$ is $O((n-m)^4 \log \beta)$, where $\beta$ is the length of the longest vector in the input basis.

## 5. Computational experiments.

The goal of the computational experiments reported in this section is threefold.

- In section 5.1 we compare the strength of $\ell$-cuts generated from different reduced bases leading to different $\boldsymbol{Q}/\boldsymbol{W}$ pairs as discussed in section 4.2.
- In section 5.2 we compare $\ell$-cuts from a single $\boldsymbol{Q}/\boldsymbol{W}$ pair (the "best" discussed in section 5.1) against standard GMIs and $k$-cuts [17]. More precisely, we considered those two families of cutting planes because the former is the standard reference for cutting plane generation, while the generation of the latter has some similarities with our approach, as previously discussed.
- In section 5.3 we compare the strength of $\ell$-cuts from a single $\boldsymbol{Q}/\boldsymbol{W}$ pair obtained by iteratively separating from the tableau, i.e., by increasing the rank, with approximate closure counterparts ($\ell$-cuts, lift-and-project, and split closures), i.e., by optimizing over the row aggregation.

Our computational investigation is focused on IPs in which all of the constraints are of knapsack type. In the context of separating $\ell$-cuts, applying the lattice reformulation to the whole matrix of such problems seems like a natural approach. Moreover, such problems are expected to be amenable to both $k$-cuts and $\ell$-cuts. An alternative would be to consider MIP problems with different types of constraints for which subsets of the constraints are treated by the lattice reformulation. Here one would first have to identify the parts of the problem (if any) on which a lattice reformulation might be effective. Identifying such substructures of general MIPs is a research project in itself and goes beyond the scope of this paper.

The test instances are obtained as in Cornuéjols et al. [17] except that the matrix coefficients $a_{ij}$, requirements $b_i$, and variable upper bounds $h_j$ are required to be integer. Specifically, the objective function coefficients $c_j$ are generated uniformly at random in [1,1000] and the coefficients $a_{ij}$ are integer-generated uniformly at random in [1,1000]. For binary instances, denoted by "B", and for instances with unbounded integer variables, denoted by "U", we compute $b_i$ as $b_i = \lfloor 0.5 \sum_{j=1}^{n} a_{ij} \rfloor$. For instances with bounded integers, denoted by "I", the $h_j$ are generated uniformly in [5,10] and $b_i = \lfloor 0.5 \sum_{j=1}^{n} h_j a_{ij} \rfloor$. We considered equality constrained instances, which are more suitable for basis reduction techniques, and we focused only on feasible problems, in order to be able to measure the integrality gap closed by $\ell$-cuts. Concerning the size of the problems, we considered instances with 1 row only ($m = 1$) and number of variables $n$ varying in $\{10, 20, 50, 100\}$, and multirow instances with $m \in \{2, 3, 4\}$ and $n = 50$. Despite the relatively small size of the instances, it is worth to remark that this kind of problems may still be challenging for state-of-the-art MIP solvers. In particular, none of the instances with $m = 4$ can be solved to optimality by CPLEX 12.7.0 [26] within 3 hours of time limit on an Intel Xeon 5160 quadcore CPUs machine running at 3.00 GHz with 8 GB RAM.

Our experiments are focused on strengthening the LP relaxation by separating cuts in a cutting plane fashion. Specifically, we compare the different types of cuts by measuring the integrality gap closed by applying 1, 5, or 10 rounds of cuttings planes to the LP relaxation of the problem. In the tables, we do not report on the computing times for reformulating the problem and for cut separation because, on the instances considered in our study, both times appear to be negligible. Concerning the cut separation, we remark that all considered cut types (GMIs, $k$-cuts, $\ell$-cuts) can be seen as "tableau" cuts, as they are separated by applying some basic algebraic operations and closed formula starting from the inverse of the optimal basis that is readily available. In our experiments, we did not observe any noticeable difference among the separation times for the different cut types, which are all a fraction of a second. Concerning the problem reformulation, performing LLL-reduction on the instances in our study takes less than a second of CPU time. Of course, the reformulation time might become an issue on larger instances. The code used in our experiments can be obtained from the authors upon request.

**5.1. Comparing the effect of basis reduction algorithms.** In this section, we examine the effect of the basis reduction method used to generate lattice basis matrix $\boldsymbol{Q}$ on the quality of the resulting $\ell$-cuts. In addition, as a reference, we compare with GMI cuts. More precisely, we consider the three reduction methods LLL-low, LLL, and KZ mentioned in section 4.2.

Table 1 reports on the results of the comparisons between GMIs from the optimal LP tableau, denoted by GMI, $\ell$-cuts from the reduction method LLL-low, denoted by $\ell$-LLL-low, $\ell$-cuts from the reduction method LLL, denoted by $\ell$-LLL, $\ell$-cuts from the reduction method KZ, denoted by $\ell$-KZ, and a combination of GMIs and $\ell$-LLL, denoted by GMI + $\ell$-LLL).

The other column headings are $R$ for the number of rounds of cuts, followed by $n$ and $m$ for the number of variables and constraints, respectively, and $T$ for the type of the instance ("B" for binary instances, "I" for instances with bounded integer variables, "U" for instances with unbounded integer variables). Then, for each approach, we report on the number of cuts generated and the percentage of the gap that is closed between the optimal LP and IP values, on average over 20 instances.

The results in Table 1 clearly show that the gap closed by $\ell$-cuts, independently of the basis reduction method, is significantly larger than that closed by only using GMIs, but the number of cuts is much larger. Moreover, by using a strongly reduced lattice basis ($\ell$-LLL or $\ell$-KZ), we obtain a significantly larger gap reduction than with a weaker reduction ($\ell$-LLL-low). The gaps closed for the $\ell$-LLL and $\ell$-KZ reductions are not significantly different, typically varying by less than 1%. As the LLL reduction is much cheaper to compute, we will just report the $\ell$-LLL results for further comparisons, although we performed the computation with both, confirming that the results are very similar.

Concerning the type of instances, we can observe that the gaps closed for unbounded integer instances are larger than those for bounded integer instances, which are in turn larger than those for binary ones. Unfortunately, as the number of rows increases from 1 to 4, the gaps closed decrease significantly, while, on the bright side, increasing the number of rounds up to 10 gives nontrivial improvements. Finally, GMIs very marginally improve on $\ell$-LLL, which somehow demonstrates that the strength of $\ell$-cuts shown by this experiment does not only depend on the number of cuts generated.

**5.2. Comparing $k$-cuts and $\ell$-cuts.** In this section, we compare the behavior of $\ell$-cuts and $k$-cuts. More precisely, we separate $k$-cuts in the following two possible ways. For each tableau row, with basic variable, say, $x_j$, we

1. multiply the row by all integer values $k = 1, \dots, 10$, and we thereby generate 10 possibly different $k$-cuts,
2. multiply the row by all integers $w_{ij}$, $i = 1, \dots, n-m$, and we generate $n-m$ possibly different $k$-cuts.

In other words, we either use "trivial" values for $k$ or **individual** $k$'s from the reduced basis LLL. Note that, for the latter, $k$-cuts and $\ell$-cuts are identical for the special case of $R = 1$ and $m = 1$; see section 4.1.

Table 2 reports on the results of the comparisons among GMIs from the optimal LP tableau, denoted by GMI, $k$-cuts of type 1 above, denoted by $k-10$, $k$-cuts of type 2 above, denoted by $k$-LLL, a combination of GMIs and $k$-cuts, denoted by GMI + $k$-LLL, $\ell$-cuts from LLL-reduced bases, denoted as before by $\ell$-LLL, and a combination of GMIs and $\ell$-LLL, denoted by GMI+$\ell$-LLL. (Note that columns GMI, $\ell$-LLL, and GMI+$\ell$-LLL are the same as in Table 1.)

The results in Table 2 clearly show that for $R > 1$ the gap closed by $\ell$-LLL is significantly larger than that closed by $k$-LLL and with far fewer cuts. Recall that the entries for $k$-LLL and $\ell$-LLL are necessarily identical for $R = 1$ and $m = 1$. Moreover, the gap closed by $k$-LLL is slightly larger than that of $k - 10$ but with more cuts in general. Finally, the improvement of GMIs + $k$-LLL with respect to $k$-LLL is much more significant than that of GMIs + $\ell$-LLL with respect to $\ell$-LLL.

**5.3. Comparing rank and row aggregation.** In this section, we compare the use of $\ell$-cuts in multiple rounds, as in the previous tables, i.e., by using for separation

TABLE 1
*Comparing how the quality of cuts depend on the basis reduction method.*

| | | | | GMI | | $\ell$-LLL-low | | $\ell$-LLL | | $\ell$-KZ | | GMI + $\ell$-LLL | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $R$ | $n$ | $m$ | T | Cuts | %gap | Cuts | %gap | Cuts | %gap | Cuts | %gap | Cuts | %gap |
| 1 | 10 | 1 | B | 1.0 | 18.61 | 5.8 | 44.19 | 9.0 | 50.86 | 9.0 | 51.21 | 9.9 | 51.17 |
| | 20 | 1 | B | 1.0 | 9.75 | 6.6 | 23.08 | 14.7 | 34.83 | 15.2 | 36.63 | 15.5 | 34.88 |
| | 50 | 1 | B | 1.0 | 12.41 | 6.3 | 23.62 | 15.1 | 34.40 | 15.1 | 33.55 | 15.4 | 34.46 |
| | 100 | 1 | B | 1.0 | 10.66 | 6.5 | 25.54 | 15.3 | 31.00 | 12.9 | 31.16 | 15.4 | 31.00 |
| | 10 | 1 | I | 1.0 | 15.67 | 6.1 | 45.63 | 8.9 | 55.10 | 8.9 | 56.62 | 9.9 | 55.23 |
| | 20 | 1 | I | 1.0 | 13.36 | 7.2 | 30.17 | 14.4 | 39.88 | 14.7 | 41.46 | 15.4 | 40.45 |
| | 50 | 1 | I | 1.0 | 11.59 | 6.6 | 20.48 | 15.5 | 35.22 | 15.3 | 32.62 | 15.6 | 35.22 |
| | 100 | 1 | I | 1.0 | 13.83 | 7.2 | 27.72 | 15.9 | 34.71 | 13.8 | 37.92 | 16.1 | 34.71 |
| | 10 | 1 | U | 1.0 | 37.27 | 5.3 | 70.12 | 7.8 | 76.65 | 7.8 | 74.23 | 8.6 | 76.65 |
| | 20 | 1 | U | 1.0 | 34.88 | 6.1 | 65.65 | 10.4 | 77.52 | 11.1 | 77.48 | 10.9 | 77.96 |
| | 50 | 1 | U | 1.0 | 62.36 | 4.5 | 85.17 | 7.5 | 92.13 | 7.5 | 89.71 | 7.6 | 92.13 |
| | 100 | 1 | U | 1.0 | 64.07 | 3.4 | 85.89 | 5.1 | 98.59 | 4.9 | 98.71 | 5.3 | 98.59 |
| | 50 | 2 | B | 2.0 | 5.31 | 10.0 | 9.30 | 26.6 | 12.03 | 31.6 | 12.32 | 27.6 | 12.13 |
| | 50 | 3 | B | 3.0 | 2.55 | 11.9 | 3.62 | 32.5 | 5.10 | 41.8 | 5.68 | 34.7 | 5.19 |
| | 50 | 4 | B | 4.0 | 1.17 | 14.8 | 1.66 | 37.0 | 2.06 | 45.4 | 2.25 | 40.4 | 2.14 |
| | 50 | 2 | U | 2.0 | 13.02 | 10.0 | 22.23 | 26.6 | 32.14 | 31.5 | 32.58 | 27.7 | 32.50 |
| | 50 | 3 | U | 3.0 | 5.48 | 11.9 | 9.34 | 32.5 | 12.26 | 41.7 | 13.60 | 34.9 | 12.68 |
| | 50 | 4 | U | 4.0 | 3.48 | 14.9 | 4.91 | 37.2 | 6.95 | 45.4 | 7.42 | 40.4 | 6.99 |
| 5 | 10 | 1 | B | 9.0 | 35.29 | 28.2 | 68.88 | 36.6 | 79.98 | 36.3 | 81.02 | 51.8 | 81.23 |
| | 20 | 1 | B | 10.4 | 25.32 | 42.2 | 47.77 | 75.3 | 57.27 | 79.6 | 57.93 | 96.1 | 57.04 |
| | 50 | 1 | B | 10.5 | 28.08 | 44.6 | 43.38 | 91.7 | 51.77 | 92.4 | 50.78 | 100.4 | 52.06 |
| | 100 | 1 | B | 10.1 | 28.91 | 45.8 | 44.23 | 95.0 | 50.46 | 87.9 | 50.38 | 99.5 | 50.58 |
| | 10 | 1 | I | 8.3 | 31.12 | 30.4 | 68.06 | 40.3 | 77.55 | 40.4 | 79.05 | 57.9 | 77.55 |
| | 20 | 1 | I | 8.0 | 25.27 | 41.4 | 51.85 | 74.0 | 65.32 | 72.6 | 64.08 | 91.9 | 67.32 |
| | 50 | 1 | I | 7.7 | 22.17 | 40.9 | 41.89 | 88.8 | 57.70 | 88.7 | 54.03 | 95.2 | 57.73 |
| | 100 | 1 | I | 7.9 | 25.27 | 46.2 | 48.96 | 93.3 | 57.45 | 87.5 | 57.45 | 97.7 | 59.02 |
| | 10 | 1 | U | 6.6 | 53.44 | 17.5 | 87.47 | 22.2 | 95.12 | 22.7 | 94.36 | 30.5 | 95.58 |
| | 20 | 1 | U | 5.7 | 57.01 | 23.2 | 90.73 | 29.5 | 97.82 | 24.1 | 99.04 | 35.0 | 98.65 |
| | 50 | 1 | U | 4.5 | 84.83 | 10.2 | 94.83 | 16.6 | 95.73 | 15.9 | 95.03 | 17.6 | 95.73 |
| | 100 | 1 | U | 4.1 | 81.91 | 8.1 | 97.67 | 7.6 | 100.00 | 6.4 | 100.00 | 8.0 | 100.00 |
| | 50 | 2 | B | 19.1 | 11.20 | 62.0 | 15.03 | 141.8 | 18.09 | 167.9 | 18.66 | 158.8 | 18.36 |
| | 50 | 3 | B | 25.5 | 4.66 | 72.1 | 5.73 | 170.4 | 7.59 | 210.9 | 8.30 | 196.4 | 7.61 |
| | 50 | 4 | B | 31.1 | 2.08 | 85.0 | 2.68 | 189.8 | 3.30 | 227.4 | 3.79 | 224.8 | 3.36 |
| | 50 | 2 | U | 15.9 | 19.87 | 59.2 | 33.04 | 139.9 | 40.10 | 162.3 | 40.98 | 156.2 | 41.17 |
| | 50 | 3 | U | 21.9 | 9.75 | 68.2 | 14.71 | 166.9 | 17.14 | 208.2 | 18.51 | 192.9 | 17.63 |
| | 50 | 4 | U | 28.2 | 5.45 | 81.7 | 6.87 | 188.7 | 9.11 | 226.9 | 9.66 | 222.0 | 9.18 |
| 10 | 10 | 1 | B | 24.6 | 42.46 | 55.7 | 73.29 | 61.2 | 82.80 | 62.1 | 84.12 | 98.0 | 83.78 |
| | 20 | 1 | B | 28.8 | 31.28 | 91.0 | 51.74 | 149.1 | 60.55 | 160.4 | 61.13 | 198.8 | 60.08 |
| | 50 | 1 | B | 30.5 | 32.59 | 100.0 | 45.48 | 192.6 | 54.40 | 194.1 | 53.39 | 214.5 | 54.65 |
| | 100 | 1 | B | 30.4 | 32.62 | 102.6 | 46.52 | 197.1 | 54.13 | 185.4 | 53.18 | 206.7 | 54.19 |
| | 10 | 1 | I | 20.6 | 38.43 | 62.2 | 71.23 | 74.8 | 80.21 | 74.7 | 81.23 | 115.9 | 80.16 |
| | 20 | 1 | I | 21.7 | 28.71 | 87.4 | 57.48 | 145.8 | 68.55 | 143.3 | 68.61 | 184.0 | 70.02 |
| | 50 | 1 | I | 19.2 | 26.97 | 88.3 | 46.20 | 184.6 | 61.12 | 184.8 | 58.64 | 201.6 | 60.96 |
| | 100 | 1 | I | 21.1 | 28.98 | 98.4 | 51.84 | 191.7 | 60.43 | 184.9 | 60.35 | 204.5 | 61.82 |
| | 10 | 1 | U | 14.2 | 58.53 | 32.9 | 91.43 | 35.8 | 95.98 | 36.4 | 95.91 | 49.9 | 96.27 |
| | 20 | 1 | U | 12.8 | 68.65 | 39.6 | 95.33 | 37.1 | 99.76 | 24.7 | 100.00 | 39.1 | 100.00 |
| | 50 | 1 | U | 9.1 | 89.69 | 13.9 | 95.55 | 19.1 | 96.52 | 21.0 | 95.99 | 20.2 | 96.52 |
| | 100 | 1 | U | 7.3 | 89.50 | 10.8 | 99.33 | 7.6 | 100.00 | 6.4 | 100.00 | 8.0 | 100.00 |
| | 50 | 2 | B | 49.2 | 12.32 | 133.8 | 16.04 | 288.7 | 18.92 | 340.3 | 19.55 | 330.7 | 19.04 |
| | 50 | 3 | B | 61.4 | 5.02 | 153.1 | 6.14 | 345.1 | 7.85 | 423.2 | 8.61 | 404.1 | 7.95 |
| | 50 | 4 | B | 71.8 | 2.27 | 179.3 | 2.87 | 383.5 | 3.43 | 455.0 | 3.93 | 463.6 | 3.47 |
| | 50 | 2 | U | 36.5 | 21.70 | 124.7 | 35.30 | 281.5 | 42.25 | 326.5 | 42.71 | 318.8 | 43.02 |
| | 50 | 3 | U | 51.0 | 10.41 | 144.0 | 15.45 | 336.5 | 17.93 | 416.7 | 19.23 | 394.2 | 18.24 |
| | 50 | 4 | U | 62.3 | 5.77 | 168.3 | 7.39 | 379.0 | 9.53 | 453.6 | 10.12 | 451.9 | 9.55 |

Table 2
*Comparing $k$-cuts and $\ell$-cuts.*

| | | | | GMI | | $k-10$ | | $k$-LLL | | GMI + $k$-LLL | | $\ell$-LLL | | GMI + $\ell$-LLL | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $R$ | $n$ | $m$ | T | Cuts | %gap | Cuts | %gap | Cuts | %gap | Cuts | %gap | Cuts | %gap | Cuts | %gap |
| 1 | 10 | 1 | B | 1.0 | 18.61 | 10.0 | 34.46 | 9.0 | 50.86 | 9.9 | 51.17 | 9.0 | 50.86 | 9.9 | 51.17 |
| | 20 | 1 | B | 1.0 | 9.75 | 10.0 | 23.74 | 14.7 | 34.83 | 15.5 | 34.88 | 14.7 | 34.83 | 15.5 | 34.88 |
| | 50 | 1 | B | 1.0 | 12.41 | 10.0 | 26.83 | 15.1 | 34.40 | 15.4 | 34.46 | 15.1 | 34.40 | 15.4 | 34.46 |
| | 100 | 1 | B | 1.0 | 10.66 | 10.0 | 29.40 | 15.3 | 31.00 | 15.4 | 31.00 | 15.3 | 31.00 | 15.4 | 31.00 |
| | 10 | 1 | I | 1.0 | 15.67 | 10.0 | 30.16 | 8.9 | 55.10 | 9.9 | 55.23 | 8.9 | 55.10 | 9.9 | 55.23 |
| | 20 | 1 | I | 1.0 | 13.36 | 10.0 | 30.92 | 14.4 | 39.88 | 15.4 | 40.45 | 14.4 | 39.88 | 15.4 | 40.45 |
| | 50 | 1 | I | 1.0 | 11.59 | 10.0 | 24.08 | 15.5 | 35.22 | 15.6 | 35.22 | 15.5 | 35.22 | 15.6 | 35.22 |
| | 100 | 1 | I | 1.0 | 13.83 | 10.0 | 31.55 | 15.9 | 34.71 | 16.1 | 34.71 | 15.9 | 34.71 | 16.1 | 34.71 |
| | 10 | 1 | U | 1.0 | 37.27 | 9.8 | 66.25 | 7.8 | 76.65 | 8.6 | 76.65 | 7.8 | 76.65 | 8.6 | 76.65 |
| | 20 | 1 | U | 1.0 | 34.88 | 9.7 | 74.18 | 10.4 | 77.52 | 10.9 | 77.96 | 10.4 | 77.52 | 10.9 | 77.96 |
| | 50 | 1 | U | 1.0 | 62.36 | 7.4 | 87.95 | 7.5 | 92.13 | 7.6 | 92.13 | 7.5 | 92.13 | 7.6 | 92.13 |
| | 100 | 1 | U | 1.0 | 64.07 | 5.6 | 98.71 | 5.1 | 98.59 | 5.3 | 98.59 | 5.1 | 98.59 | 5.3 | 98.59 |
| | 50 | 2 | B | 2.0 | 5.31 | 19.7 | 10.11 | 50.0 | 12.87 | 50.9 | 12.87 | 26.6 | 12.03 | 27.6 | 12.13 |
| | 50 | 3 | B | 3.0 | 2.55 | 30.0 | 4.22 | 92.5 | 5.54 | 94.5 | 5.61 | 32.5 | 5.10 | 34.7 | 5.19 |
| | 50 | 4 | B | 4.0 | 1.17 | 40.0 | 2.17 | 139.7 | 2.51 | 142.9 | 2.53 | 37.0 | 2.06 | 40.4 | 2.14 |
| | 50 | 2 | U | 2.0 | 13.02 | 20.0 | 24.49 | 42.7 | 33.06 | 43.5 | 33.14 | 26.6 | 32.14 | 27.7 | 32.50 |
| | 50 | 3 | U | 3.0 | 5.48 | 30.0 | 11.20 | 87.9 | 13.41 | 89.9 | 13.58 | 32.5 | 12.26 | 34.9 | 12.68 |
| | 50 | 4 | U | 4.0 | 3.48 | 40.0 | 6.32 | 137.2 | 8.50 | 139.9 | 8.50 | 37.2 | 6.95 | 40.4 | 6.99 |
| 5 | 10 | 1 | B | 9.0 | 35.29 | 127.6 | 47.98 | 105.7 | 54.14 | 126.1 | 62.70 | 36.6 | 79.98 | 51.8 | 81.23 |
| | 20 | 1 | B | 10.4 | 25.32 | 156.1 | 33.03 | 235.8 | 37.52 | 266.2 | 45.44 | 75.3 | 57.27 | 96.1 | 57.04 |
| | 50 | 1 | B | 10.5 | 28.08 | 157.8 | 38.24 | 283.1 | 41.54 | 300.0 | 42.91 | 91.7 | 51.77 | 100.4 | 52.06 |
| | 100 | 1 | B | 10.1 | 28.91 | 164.0 | 40.02 | 261.0 | 40.92 | 270.3 | 41.71 | 95.0 | 50.46 | 99.5 | 50.58 |
| | 10 | 1 | I | 8.3 | 31.12 | 122.9 | 42.23 | 140.5 | 56.14 | 145.2 | 64.05 | 40.3 | 77.55 | 57.9 | 77.55 |
| | 20 | 1 | I | 8.0 | 25.27 | 135.1 | 36.05 | 234.9 | 43.39 | 256.6 | 49.93 | 74.0 | 65.32 | 91.9 | 67.32 |
| | 50 | 1 | I | 7.7 | 22.17 | 132.9 | 32.33 | 236.3 | 42.28 | 243.4 | 44.61 | 88.8 | 57.70 | 95.2 | 57.73 |
| | 100 | 1 | I | 7.9 | 25.27 | 138.9 | 39.05 | 254.2 | 42.27 | 262.0 | 43.13 | 93.3 | 57.45 | 97.7 | 59.02 |
| | 10 | 1 | U | 6.6 | 53.44 | 88.9 | 73.83 | 68.2 | 79.29 | 68.2 | 83.30 | 22.2 | 95.12 | 30.5 | 95.58 |
| | 20 | 1 | U | 5.7 | 57.01 | 77.8 | 84.04 | 98.7 | 81.13 | 103.0 | 87.07 | 29.5 | 97.82 | 35.0 | 98.65 |
| | 50 | 1 | U | 4.5 | 84.83 | 30.9 | 91.36 | 43.9 | 94.87 | 45.0 | 94.88 | 16.6 | 95.73 | 17.6 | 95.73 |
| | 100 | 1 | U | 4.1 | 81.91 | 12.8 | 100.00 | 16.2 | 100.00 | 15.9 | 100.00 | 7.6 | 100.00 | 8.0 | 100.00 |
| | 50 | 2 | B | 19.1 | 11.20 | 244.3 | 13.47 | 675.0 | 15.27 | 711.1 | 15.98 | 141.8 | 18.09 | 158.8 | 18.36 |
| | 50 | 3 | B | 25.5 | 4.66 | 320.6 | 5.80 | 1072.4 | 6.43 | 1104.2 | 6.79 | 170.4 | 7.59 | 196.4 | 7.61 |
| | 50 | 4 | B | 31.1 | 2.08 | 372.8 | 2.79 | 1439.8 | 2.74 | 1472.2 | 2.97 | 189.8 | 3.30 | 224.8 | 3.36 |
| | 50 | 2 | U | 15.9 | 19.87 | 225.0 | 28.63 | 654.0 | 36.15 | 663.0 | 36.99 | 139.9 | 40.10 | 156.2 | 41.17 |
| | 50 | 3 | U | 21.9 | 9.75 | 314.9 | 13.29 | 1032.3 | 14.76 | 1031.1 | 15.97 | 166.9 | 17.14 | 192.9 | 17.63 |
| | 50 | 4 | U | 28.2 | 5.45 | 358.6 | 7.39 | 1432.1 | 8.93 | 1495.8 | 9.51 | 188.7 | 9.11 | 222.0 | 9.18 |
| 10 | 10 | 1 | B | 24.6 | 42.46 | 301.9 | 52.21 | 223.9 | 54.93 | 300.7 | 65.92 | 61.2 | 82.80 | 98.0 | 83.78 |
| | 20 | 1 | B | 28.8 | 31.28 | 359.9 | 37.12 | 510.0 | 38.10 | 645.3 | 47.84 | 149.1 | 60.55 | 198.8 | 60.08 |
| | 50 | 1 | B | 30.5 | 32.59 | 397.3 | 40.99 | 637.5 | 43.25 | 688.2 | 45.17 | 192.6 | 54.40 | 214.5 | 54.65 |
| | 100 | 1 | B | 30.4 | 32.62 | 398.8 | 42.82 | 600.7 | 44.02 | 624.3 | 44.82 | 197.1 | 54.13 | 206.7 | 54.19 |
| | 10 | 1 | I | 20.6 | 38.43 | 269.6 | 44.70 | 285.8 | 56.77 | 318.5 | 65.69 | 74.8 | 80.21 | 115.9 | 80.16 |
| | 20 | 1 | I | 21.7 | 28.71 | 308.3 | 37.71 | 499.9 | 44.22 | 551.8 | 52.77 | 145.8 | 68.55 | 184.0 | 70.02 |
| | 50 | 1 | I | 19.2 | 26.97 | 320.5 | 34.75 | 515.2 | 43.31 | 532.9 | 46.65 | 184.6 | 61.12 | 201.6 | 60.96 |
| | 100 | 1 | I | 21.1 | 28.98 | 326.8 | 43.72 | 536.0 | 43.22 | 541.4 | 45.74 | 191.7 | 60.43 | 204.5 | 61.82 |
| | 10 | 1 | U | 14.2 | 58.53 | 183.3 | 76.06 | 143.6 | 80.54 | 143.3 | 85.03 | 35.8 | 95.98 | 49.9 | 96.27 |
| | 20 | 1 | U | 12.8 | 68.65 | 158.7 | 87.45 | 212.3 | 82.67 | 212.2 | 89.62 | 37.1 | 99.76 | 39.1 | 100.00 |
| | 50 | 1 | U | 9.1 | 89.69 | 63.3 | 92.12 | 77.9 | 94.98 | 75.7 | 95.04 | 19.1 | 96.52 | 20.2 | 96.52 |
| | 100 | 1 | U | 7.3 | 89.50 | 12.8 | 100.00 | 16.2 | 100.00 | 15.9 | 100.00 | 7.6 | 100.00 | 8.0 | 100.00 |
| | 50 | 2 | B | 49.2 | 12.32 | 573.2 | 14.08 | 1486.7 | 15.69 | 1580.3 | 16.62 | 288.7 | 18.92 | 330.7 | 19.04 |
| | 50 | 3 | B | 61.4 | 5.02 | 726.6 | 6.05 | 2227.4 | 6.64 | 2318.9 | 6.98 | 345.1 | 7.85 | 404.1 | 7.95 |
| | 50 | 4 | B | 71.8 | 2.27 | 847.6 | 2.90 | 2895.1 | 2.84 | 3035.1 | 3.07 | 383.5 | 3.43 | 463.6 | 3.47 |
| | 50 | 2 | U | 36.5 | 21.70 | 486.8 | 29.71 | 1323.9 | 36.65 | 1360.1 | 38.22 | 281.5 | 42.25 | 318.8 | 43.02 |
| | 50 | 3 | U | 51.0 | 10.41 | 672.9 | 13.75 | 2102.6 | 14.95 | 2118.5 | 16.26 | 336.5 | 17.93 | 394.2 | 18.24 |
| | 50 | 4 | U | 62.3 | 5.77 | 760.1 | 7.64 | 2812.4 | 9.00 | 2960.1 | 9.65 | 379.0 | 9.53 | 451.9 | 9.55 |

the row aggregation provided by the simplex algorithm, with the case in which we optimize over the aggregation by solving an LP but we stay at rank 1; i.e., we only use the original constraints and the $\boldsymbol{W}$-matrix. The latter procedure, if iterated, allows us to compute the *approximated strengthened* $\ell$-LLL *closure* by adapting the algorithm proposed by Bonami [9] for the *strengthened lift-and-project closure*. More precisely,

- The strengthened lift-and-project closure of a mixed-integer linear program is the polyhedron obtained by intersecting all strengthened lift-and-project cuts [6, 22] obtained from its initial formulation or equivalently all GMIs read from all tableaus corresponding to feasible and infeasible bases of the LP relaxation. An approximation of this closure is computed by iteratively generating lift-and-project cuts and strengthening them by integer lifting; see [9].
- Analogously, given a reduced $\boldsymbol{W}$-matrix to generate rank-1 $\ell$-cuts, the approximated strengthened $\ell$-LLL closure is computed as follows. If $\boldsymbol{x}^*$ is the optimal LP solution and $\boldsymbol{w}^i x^* \notin \mathbb{Z}$, one generates an intersection cut [6] on the disjunction, $\boldsymbol{w}^i\boldsymbol{x} \le \lfloor \boldsymbol{w}^i\boldsymbol{x}^* \rfloor$ and $\boldsymbol{w}^i\boldsymbol{x} \ge \lceil \boldsymbol{w}^i\boldsymbol{x}^* \rceil$, which is then strengthened. This is repeated for each row $i$ of $\boldsymbol{W}$ at each iteration until no more violated cuts are found.

In terms of closures, the comparison is completed by reporting on the results for the split closure. Exploiting the result reported [21] that shows the equivalence between the split closure and the mixed-integer rounding (MIR) closure, the split closure is computed by iteratively separating violated MIR cuts through the solution of a mixed-integer program as in [21].

Table 3 reports on the results on the comparisons between

- 10 rounds of (a) $\ell$-LLL cuts, (b) a combination of GMIs and $\ell$-LLL cuts, denoted by GMI+$\ell$-LLL, and
- the approximated closures of (c) strengthened lift-and-project cuts, denoted by "str. L&P," (d) strengthened $\ell$-LLL cuts, denoted by "str. $\ell$-LLL," (e) split cuts, denoted by "split."

In contrast to the cases of strengthened L&P and $\ell$-LLL closures, the term "approximated" for the split closure refers to the fact that the computation is stopped after a time limit of 5 hours. Such a time limit affects only the multirow instances with binary variables and this is indicated in the table by "*."

The results in Table 3 clearly show that growing the rank of the $\ell$-cuts gives generally better results than optimizing over the approximate closure of the disjunctions in the $\boldsymbol{W}$-matrix although there is no domination. Nevertheless, it is confirmed that the approximated strengthened $\ell$-LLL closure is way stronger than the approximated strengthened L&P closure. In other words, elementary disjunctions in the reformulated space are stronger than elementary disjunctions in the original space. With few exceptions, neither the strengthened $\ell$-LLL closure nor the strengthened L&P closure provide a good approximation of the rank-1 split closure. Finally, separating both $\ell$-LLL and L&P cuts together does not significantly improve over $\ell$-LLL alone, although the results are not explicitly reported in the table.

**6. Concluding remarks.** Our $\ell$-cuts are generated based on general disjunctions originating from information on the lattice structure of the underlying problem. For the test instances, which are similar to the instances used by [17] in their computational study of $k$-cuts, we observe that the lattice structure gives useful information to obtain cuts that improve on standard GMI/split cuts and $k$-cuts. For single-row problems, a large percentage of the integrality gap is closed. For multirow problems the results are not as good, and it remains a challenge to identify cuts that can be

TABLE 3
*Comparing higher rank cuts with rank-1 closures.*

| | | | 10 rounds | | | | "Approximated" closures | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $\ell$-LLL | | GMI + $\ell$-LLL | | str. L&P | | str. $\ell$-LLL | | split | | |
| $n$ | $m$ | T | Cuts | %gap | Cuts | %gap | Cuts | %gap | Cuts | %gap | Cuts | %gap | |
| 10 | 1 | B | 61.2 | 82.80 | 98.0 | 83.78 | 4.1 | 29.87 | 41.6 | 99.73 | 62.3 | 100.00 | |
| 20 | 1 | B | 149.1 | 60.55 | 198.8 | 60.08 | 4.0 | 18.56 | 71.4 | 61.93 | 148.4 | 84.11 | |
| 50 | 1 | B | 192.6 | 54.40 | 214.5 | 54.65 | 4.1 | 22.32 | 48.0 | 50.84 | 174.7 | 88.29 | |
| 100 | 1 | B | 197.1 | 54.13 | 206.7 | 54.19 | 4.4 | 21.38 | 42.8 | 46.81 | 162.3 | 89.72 | |
| 10 | 1 | I | 74.8 | 80.21 | 115.9 | 80.16 | 1.3 | 16.89 | 14.3 | 63.87 | 48.3 | 88.97 | |
| 20 | 1 | I | 145.8 | 68.55 | 184.0 | 70.02 | 1.3 | 13.68 | 21.2 | 47.42 | 53.9 | 81.90 | |
| 50 | 1 | I | 184.6 | 61.12 | 201.6 | 60.96 | 1.2 | 12.26 | 21.1 | 42.02 | 61.6 | 82.28 | |
| 100 | 1 | I | 191.7 | 60.43 | 204.5 | 61.82 | 1.3 | 14.94 | 20.2 | 42.28 | 57.0 | 85.22 | |
| 10 | 1 | U | 35.8 | 95.98 | 49.9 | 96.27 | 1.0 | 37.27 | 8.8 | 79.31 | 25.9 | 97.58 | |
| 20 | 1 | U | 37.1 | 99.76 | 39.1 | 100.00 | 1.0 | 34.88 | 10.4 | 77.52 | 27.0 | 92.57 | |
| 50 | 1 | U | 19.1 | 96.52 | 20.2 | 96.52 | 1.0 | 62.36 | 7.5 | 92.13 | 38.1 | 99.97 | |
| 100 | 1 | U | 7.6 | 100.00 | 8.0 | 100.00 | 1.0 | 64.07 | 5.1 | 98.59 | 35.9 | 98.70 | |
| 50 | 2 | B | 288.7 | 18.92 | 330.7 | 19.04 | 10.1 | 9.95 | 84.4 | 18.96 | 460.8 | 41.24 | * |
| 50 | 3 | B | 345.1 | 7.85 | 404.1 | 7.95 | 15.8 | 4.23 | 99.7 | 8.09 | 519.0 | 18.95 | * |
| 50 | 4 | B | 383.5 | 3.43 | 463.6 | 3.47 | 20.7 | 1.99 | 140.5 | 4.05 | 518.6 | 8.49 | * |
| 50 | 2 | U | 281.5 | 42.25 | 318.8 | 43.02 | 2.0 | 13.02 | 26.7 | 32.28 | 178.6 | 71.87 | |
| 50 | 3 | U | 336.5 | 17.93 | 394.2 | 18.24 | 3.3 | 5.57 | 34.7 | 12.81 | 342.4 | 42.08 | |
| 50 | 4 | U | 379.0 | 9.53 | 451.9 | 9.55 | 4.7 | 3.76 | 40.9 | 7.49 | 372.8 | 22.41 | |

generated within reasonable computing time and that work well on multirow problems.

We observe that the better the quality of the basis generating the lattice, the better the quality of the resulting $\ell$-cuts. We have, however, only tried one lattice reformulation [3], and given the partial success of the approach it would be useful to investigate other reformulations, in particular a reformulation that captures multirow problems better. Also, extending our approach to mixed-integer problems with different types of constraints would be interesting. Finally, we have applied our cutting planes only at the root node, i.e., to strengthen *a priori* the initial LP relaxation. This turned out not to be enough to transform instances that are very difficult for branch and bound into "easy," i.e., solvable in short computing times. A more complete approach would require to perform cut selection and, likely, to generate cuts in the tree. This is left for future research.

## REFERENCES

[1] K. Aardal, R. E. Bixby, C. A. J. Hurkens, A. K. Lenstra, and J. W. Smeltink, *Market split and basis reduction: Towards a solution of the Cornuéjols-Dawande instances*, INFORMS J. Comput., 12 (2000), pp. 192–202, https://doi.org/10.1287/ijoc.12.3.192.12635.

[2] K. Aardal and F. Eisenbrand, *Integer programming, lattices, and results in fixed dimension*, in Discrete Optimization, K. Aardal, G. L. Nemhauser, R. Weismantel, eds., Handbooks in Operations Research and Management Science 12, Elsevier, Amsterdam, 2005, pp. 171–243.

[3] K. Aardal, C. A. J. Hurkens, and A. K. Lenstra, *Solving a system of linear Diophantine equations with lower and upper bounds on the variables*, Math. Oper. Res., 25 (2000), pp. 427–442, https://doi.org/10.1287/moor.25.3.427.12219.

[4] K. Aardal and A. K. Lenstra, *Hard equality constrained integer knapsacks*, Math. Oper. Res., 29 (2004), pp. 724–738, https://doi.org/10.1287/moor.1040.0099. Erratum: Math. Oper. Res., 31 (2006), p. 846.

[5] K. Aardal and L. A. Wolsey, *Lattice based extended formulations for integer linear equality systems*, Math. Program., 121 (2010), pp. 337–352, https://doi.org/10.1007/s10107-008-0236-7.
[6] E. Balas, *Intersection cuts—a new type of cutting planes for integer programming*, Operations Res., 19 (1971), pp. 19–39, https://doi.org/10.1287/opre.19.1.19.
[7] E. Balas and A. Saxena, *Optimizing over the split closure*, Math. Program., 113 (2008), pp. 219–240, https://doi.org/10.1007/s10107-006-0049-5.
[8] R. E. Bixby, S. Ceria, C. M. McZeal, and M. W. Savelsbergh, *MIPLIB* 3.0, https://www.caam.rice.edu/bixby/miplib/miplib.html, 1996.
[9] P. Bonami, *On optimizing over lift-and-project closures*, Math. Program. Comput., 4 (2012), pp. 151–179, https://doi.org/10.1007/s12532-012-0037-0.
[10] P. Bonami, G. Cornuéjols, S. Dash, M. Fischetti, and A. Lodi, *Projected Chvátal-Gomory cuts for mixed integer linear programs*, Math. Program., 113 (2008), pp. 241–257, https://doi.org/10.1007/s10107-006-0051-y.
[11] A. Caprara and A. N. Letchford, *On the separation of split cuts and related inequalities*, Math. Program., 94 (2003), pp. 279–294, https://doi.org/10.1007/s10107-002-0320-3.
[12] V. Chvátal, *Edmonds polytopes and a hierarchy of combinatorial problems*, Discrete Math., 4 (1973), pp. 305–337, https://doi.org/10.1016/0012-365X(73)90167-2.
[13] M. Conforti, G. Cornuéjols, and G. Zambelli, *Integer Programming*, Graduate Texts in Mathematics 271, Springer, Cham, 2014, https://doi.org/10.1007/978-3-319-11008-0.
[14] W. Cook, R. Kannan, and A. Schrijver, *Chvátal closures for mixed integer programming problems*, Math. Program., 47 (1990), pp. 155–174, https://doi.org/10.1007/BF01580858.
[15] W. Cook, T. Rutherford, H. E. Scarf, and S. David, *An implementation of the generalized basis reduction algorithm for integer programming*, ORSA J. Comput., 5 (1993), pp. 206–212.
[16] G. Cornuéjols, *Valid inequalities for mixed integer linear programs*, Math. Program., 112 (2008), pp. 3–44, https://doi.org/10.1007/s10107-006-0086-0.
[17] G. Cornuéjols, Y. Li, and D. Vandenbussche, *k-cuts: A variation of Gomory mixed integer cuts from the LP tableau*, INFORMS J. Comput., 15 (2003), pp. 385–396, https://doi.org/10.1287/ijoc.15.4.385.24893.
[18] G. Cornuéjols and G. Nannicini, *Practical strategies for generating rank*-1 *split cuts in mixed-integer linear programming*, Math. Program. Comput., 3 (2011), pp. 281–318, https://doi.org/10.1007/s12532-011-0028-6.
[19] S. Dash and M. Goycoolea, *A heuristic to generate rank*-1 *GMI cuts*, Math. Program. Comput., 2 (2010), pp. 231–257, https://doi.org/10.1007/s12532-010-0018-0.
[20] S. Dash, O. Günlük, and R. Hildebrand, *Binary extended formulations of polyhedral mixed-integer sets*, Math. Program., 170 (2018), pp. 207–236, https://doi.org/10.1007/s10107-018-1294-0.
[21] S. Dash, O. Günlük, and A. Lodi, *MIR closures of polyhedral sets*, Math. Program., 121 (2010), pp. 33–60, https://doi.org/10.1007/s10107-008-0225-x.
[22] M. Fischetti, A. Lodi, and A. Tramontani, *On the separation of disjunctive cuts*, Math. Program., 128 (2011), pp. 205–230, https://doi.org/10.1007/s10107-009-0300-y.
[23] M. Fischetti and D. Salvagnin, *A relax-and-cut framework for Gomory mixed-integer cuts*, Math. Program. Comput., 3 (2011), pp. 79–102, https://doi.org/10.1007/s12532-011-0024-x.
[24] R. E. Gomory, *An algorithm for integer solutions to linear programs*, in Recent Advances in Mathematical Programming, McGraw-Hill, New York, 1963, pp. 269–302.
[25] R. E. Gomory, *Some polyhedra related to combinatorial problems*, Linear Algebra Appl., 2 (1969), pp. 451–558, https://doi.org/10.1016/0024-3795(69)90017-2.
[26] IBM CPLEX Optimizer, https://www.ibm.com/analytics/cplex-optimizer.
[27] A. Korkine and G. Zolotareff, *Sur les formes quadratiques*, Math. Ann., 6 (1873), pp. 366–389, https://doi.org/10.1007/BF01442795.
[28] B. Krishnamoorthy and G. Pataki, *Column basis reduction and decomposable knapsack problems*, Discrete Optim., 6 (2009), pp. 42–270.
[29] J. C. Lagarias, H. W. Lenstra, Jr., and C.-P. Schnorr, *Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal lattice*, Combinatorica, 10 (1990), pp. 333–348, https://doi.org/10.1007/BF02128669.
[30] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász, *Factoring polynomials with rational coefficients*, Math. Ann., 261 (1982), pp. 515–534, https://doi.org/10.1007/BF01457454.
[31] H. W. Lenstra Jr., *Integer programming with a fixed number of variables*, Math. Oper. Res., 8 (1983), pp. 538–548.

[32] Q. Louveaux and L. A. Wolsey, *Combining problem structure with basis reduction to solve a class of har integer programs*, Math. Oper. Res., 27 (2002), pp. 470–484.

[33] S. Mehrotra and Z. Li, *Branching on hyperplane methods for mixed integer linear and convex programming using adjoint lattices*, J. Global Optim., 49 (2011), pp. 623–649, https://doi.org/10.1007/s10898-010-9554-4.

[34] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, Inc., New York, 1988, https://doi.org/10.1002/9781118627372.

[35] G. L. Nemhauser and L. A. Wolsey, *A recursive procedure to generate all cuts for* 0-1 *mixed integer programs*, Math. Program., 46 (1990), pp. 379–390, https://doi.org/10.1007/BF01585752.

[36] C.-P. Schnorr, *A hierarchy of polynomial time lattice basis reduction algorithms*, Theoret. Comput. Sci., 53 (1987), pp. 201–224.

[37] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley-Interscience Series in Discrete Mathematics, John Wiley & Sons Ltd., Chichester, 1986.

[38] V. Shoup, *NTL: A library for doing Number Theory*, https://www.shoup.net/ntl/.