CAN CUT GENERATING FUNCTIONS BE GOOD *AND* EFFICIENT?*

AMITABH BASU[†] AND SRIRAM SANKARANARAYANAN [‡]

Abstract. Making cut generating functions (CGFs) computationally viable is a central question in modern integer programming research. One would like to find CGFs that are simultaneously *good*, i.e., there are good guarantees for the cutting planes they generate, and *efficient*, meaning that the values of the CGFs can be computed cheaply (with procedures that have some hope of being implemented in current solvers). We investigate in this paper to what extent this balance can be struck. We propose a family of CGFs which, in a sense, achieves this harmony between *good* and *efficient*. In particular, we provide a parameterized family of $b+\mathbb{Z}^n$ free sets to derive CGFs from and show that our proposed CGFs give a good approximation of the closure given by CGFs obtained from all maximal $b+\mathbb{Z}^n$ free sets and their so-called *trivial liftings*, and simultaneously, show that these CGFs can be computed with explicit, efficient procedures. We provide a constructive framework to identify these sets as well as computing their trivial lifting. We follow it up with computational experiments to demonstrate this and to evaluate their practical use. Our proposed family of cuts seem to give some tangible improvement on randomly generated instances compared to GMI cuts; however, in MIPLIB 3.0 instances, and vertex cover and stable problems on random graph instances, their performance is poor.

Key words. Integer programming, Multi-row cuts, Lattice-free convex sets, Cutting planes

AMS subject classifications. 90C10, 90C11, 90C57

1. Introduction.

In this paper, we study the inequality description of sets of the form

$$X(R,P) \quad := \quad \operatorname{conv}\left\{(s,y)\in\mathbb{R}^k_+\times\mathbb{Z}^\ell_+ \mid Rs+Py\in b+\mathbb{Z}^n\right\} \tag{1.1}$$

where $n,k,\ell\in\mathbb{N}$, $R\in\mathbb{R}^{n\times k}$, $P\in\mathbb{R}^{n\times\ell}$, $b\in\mathbb{R}^n\setminus\mathbb{Z}^n$. Such sets have been the focus of intense study in the last decade, and are typically refereed to as *mixed-integer corner polyhedra* in the literature; see the surveys [26, 10, 15, 16] and [22, Chapter 6], and the references therein. One of the focal points in this recent activity has been the revival of the *cut generating function* approach, originally pioneered by Gomory and Johnson in their seminal work in the 1970s [35, 36, 38]. The phrase "cut generating function" was invented relatively recently by the authors of [19].

Definition 1.1 (Valid pair). *Fix $n\in\mathbb{N}$. A pair of real valued functions (ψ,π) on $\mathbb{R}^n$ are said to be a* valid pair *if*

$$\sum_{i=1}^{k}\psi(r_i)s_i+\sum_{i=1}^{\ell}\pi(p_i)y_i \quad \geq \quad 1 \tag{1.2}$$

is a valid inequality for $X(R,P)$ for all k,ℓ,R,P, where r_i and p_i refer to the columns of R and P respectively.

The important thing to note is that a valid pair of functions only depends on the dimension n and b, and should work for any matrices R,P with n rows, and an arbitrary number of columns. Gomory and Johnson made the discovery that not

---

*Submitted to the editors.

Funding: Both the authors gratefully acknowledge the support from NSF grant CMMI1452820

[†]Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD (abasu9@jhu.edu).

[‡]Department of Civil Engineering, Johns Hopkins University, Baltimore, MD (srirams@jhu.edu).

only do such valid pairs of functions exist, they give a unifying framework for many cut generating procedures extensively used in the integer programming community. Gomory's original motivation [34] was to choose n rows from the optimal simplex tableaux of a general mixed-integer optimization problem and apply these cut generating functions (for this particular choice of n rows of the tableaux) to obtain cutting planes for the original problem. The modern trend has been to build a more computationally tractable viewpoint of this theory. This has been possible by drawing upon novel insights into cutting plane theory by Balas from the 1970s, which was termed by him as the theory of *intersection cuts* [4]. We summarize this approach to cut generating functions next.

Given a convex set C with the origin in its interior, the *gauge function* is defined as $\psi_C(x) := \inf_{\lambda>0} \left\{\lambda : \frac{x}{\lambda} \in C\right\}$. Let S be any closed subset of $\mathbb{R}^n \setminus \{0\}$ (not necessarily convex). A closed convex set B containing 0 in its interior is said to be an *S-free convex neighborhood of* 0 if $\operatorname{int}(B) \cap S = \emptyset$. It is said to be a *maximal* S-free convex neighborhood of 0 if it is not strictly contained in another S-free convex neighborhood of 0. For brevity, we will often refer to such sets as (maximal) S-free convex sets. In this paper, we will be concerned with $S = b + \mathbb{Z}^n$, where $b \in \mathbb{R}^n \setminus \mathbb{Z}^n$. The starting point of combining Balas' intersection cuts and Gomory-Johnson's cut generating function theory is the observation that setting $\psi = \pi = \psi_B$, where $B \subseteq \mathbb{R}^n$ is a maximal $b + \mathbb{Z}^n$ free set gives a valid pair. Thus, for every maximal $b + \mathbb{Z}^n$ free set $B \subseteq \mathbb{R}^n$, we obtain a valid inequality $\sum_{i=1}^{k} \psi_B(r_i)s_i + \sum_{i=1}^{\ell} \psi_B(p_i)y_i \geq 1$ for $X(R, P)$, for all k, ℓ, R, P. Such inequalities can be implemented in a cut generating procedure in any modern solver, as long as one has a way of computing $\psi_B(r)$ efficiently, for any $r \in \mathbb{R}^n$. Here, a new ingredient has been added by modern research, which uses a result of Lovasz [41] (later refined by others) stating that all maximal $b + \mathbb{Z}^n$ free sets are polyhedra that can be written in the form $B := \{x \in \mathbb{R}^n : a^i \cdot x \leq 1, \quad i = 1, \ldots, m\}$, where $a^i \in \mathbb{R}^n$. It turns out that the gauge function of such a set is simply $\psi_B(r) = \max_{i=1}^{m} a^i \cdot r$. This now makes the computation of the coefficients of the cut $\sum_{i=1}^{k} \psi_B(r_i)s_i + \sum_{i=1}^{\ell} \psi_B(p_i)y_i \geq 1$ more concrete, compared to the original theory of Gomory and Johnson.

The next ingredient in the modern approach to cut generating functions is to use an idea due to Balas and Jeroslow [5], which they termed *monoidal strengthening*. In our context, the observation translates to the fact that one can improve the coefficients of the y_i variables, by using the integrality constraint on these variables.

Definition 1.2 (Trivial lifting). *Let $b \in \mathbb{R}^n \setminus \mathbb{Z}^n$ and let B be a maximal $(b+\mathbb{Z}^n)$-free convex set. The* trivial lifting *of $\psi_B(x)$ is defined by*

$$\widetilde{\psi_B}(x) \quad = \quad \min\left(1, \inf_{z\in\mathbb{Z}^n} \psi_B(x+z)\right) \tag{1.3}$$

One of the main outcomes of the recent computational perspective on cut generating functions can be summarized as follows [29, 35].

Theorem 1.3. *Let $b \in \mathbb{R}^n \setminus \mathbb{Z}^n$ and let B be a maximal $(b + \mathbb{Z}^n)$-free convex set. Then $(\psi_B, \widetilde{\psi_B})$ is a valid pair.*

It is important to note that given a maximal $b + \mathbb{Z}^n$ free set B, there may exist several functions $\pi : \mathbb{R}^n \to \mathbb{R}$ such that (ψ_B, π) is a valid pair; all such functions π are called *liftings* of ψ_B. The trivial lifting is only one such function. Since the variables y are nonnegative, if we have two liftings $\pi_1 \leq \pi_2$, then the cutting plane (1.2) derived from π_2 is dominated by the one derived from π_1. Thus, ideally, one would like to work with *minimal liftings*, i.e., liftings π such that there does not exist a different lifting $\pi' \neq \pi$ with $\pi' \leq \pi$. In general, the trivial lifting may not be minimal; characterizing situations when it is indeed minimal has received a lot of attention [2, 11, 17, 29, 28, 9, 21]. In fact, the trivial lifting is always an upper bound on any minimal lifting, i.e., $\pi \leq \widetilde{\psi}$ for any minimal lifting π of ψ. Thus, when the trivial lifting is minimal, it is the unique minimal lifting.

In our opinion, there are two key obstacles to implementing such cut generating functions in state-of-the-art software:

1. There are too many (in fact, infinitely many) maximal $b + \mathbb{Z}^n$ free sets to choose from. This is the problem of *cut selection*.
2. For maximal $b+\mathbb{Z}^n$ free polyhedra with complicated combinatorial structure, the computation of the trivial lifting via (1.3) is extremely challenging. Moreover, computing the values of minimal liftings, especially if the trivial lifting is not the unique minimal lifting is even more elusive, with no formulas like (1.3) available.

Thus, a central question in making cut generating function theory computationally viable, which also motivates the title of this paper, is the following.

Question 1.4. Find a "simple" subset of maximal $b+\mathbb{Z}^n$ free polyhedra such that two goals are simultaneously achieved:

(i) provide guarantees that this "simple" subset of $b + \mathbb{Z}^n$ free sets gives a good approximation of the closure obtained by throwing in cuts from all possible maximal $b + \mathbb{Z}^n$ free sets, and

(ii) cutting planes like (1.2) can be derived from them with relatively light computational overhead, either via trivial liftings or other lifting techniques.

1.1. Summary of results. The goal of this paper is to make some progress in Question 1.4. In our opinion, these results provide both theoretical evidence for the utility of cut generating functions and algorithms that are efficient enough be implemented in practice.

1. One may wonder if the trivial lifting function of the gauge can approximate any minimal lifting up to some factor. We show that there exist maximal $b+\mathbb{Z}^n$ free sets whose gauge functions have minimal liftings that are arbitrarily better than the trivial lifting (on some subset of vectors) [recall that any minimal lifting is pointwise smaller than the trivial lifting]. More formally, we establish

   THEOREM 1.5. *Let n be any natural number and $\varepsilon > 0$. There exists $b \in \mathbb{R}^n \setminus \mathbb{Z}^n$ and a family $\mathcal{F}$ of maximal $(b+\mathbb{Z}^n)$-free sets such that for any $B \in \mathcal{F}$, there exists a minimal lifting π of ψ_B and $p \in \mathbb{R}^n$ satisfying $\frac{\pi(p)}{\widetilde{\psi}_B(p)} < \varepsilon$.*

2. Given an arbitrary maximal $b + \mathbb{Z}^n$ free set B, computing the trivial lifting using (1.3) can be computationally very hard because it is equivalent to the notorious closest lattice vector problem in the algorithmic geometry of numbers literature [30]. One could potentially write an integer linear program to solve it, but this somewhat defeats the purpose of cut generating functions: one would like to compute the coefficients much faster than solving complicated optimization problems like (1.3) (and even harder IPs for general lifting). To overcome this issue, we isolate a particular family of maximal $b + \mathbb{Z}^n$ free sets that we call *generalized cross-polyhedra* (see Definition 2.1 for a precise

definition) and give an algorithm for computing the trivial lifting function for any member of this family without using a high dimensional integer linear program. For this family, one needs $O(2^n)$ time to compute the gauge function because the $b+\mathbb{Z}^n$ free sets have 2^n facets, and one needs an additional $O(n2^n)$ time to compute the trivial lifting coefficient. *Recall that n corresponds to the number of rows used to generate the cuts.* This is much better complexity compared to solving (1.3) using an integer program or a closest lattice vector (the latter will have to deal with an asymmetric, polyhedral gauge which is challenging). This is described in section 3; see Algorithm 3.1. For a subfamily of generalized cross-polyhedra, both of these computations (gauge values and trivial lifting values) can actually be done in $O(n)$ time, which we exploit in our computational tests (see subsection 4.1.2). We envision using this in software and computations in the regime $n \leq 15$. *To the best of our knowledge, no previous work provides a comparable lifting procedure that can be easily coded in software and that works for any number of rows n, even for a restricted class of $b + \mathbb{Z}^n$ free sets. Previous work on lifting that can be readily translated to code, without solving an intermediate IP, has focused on the $n = 1, 2$ case (see the relevant literature discussed below).*

3. From a theoretical perspective, we also show that our family of generalized cross-polyhedra can provide a finite approximation for the closure of cutting planes of the form

$$\sum_{i=1}^{k} \psi_B(r_i)s_i + \sum_{i=1}^{\ell} \widetilde{\psi_B}(p_i)y_i \geq 1.$$

More precisely, for any matrices $R \in \mathbb{R}^{n\times k}$, $P \in \mathbb{R}^{n\times \ell}$, and any maximal $b+\mathbb{Z}^n$ free set B, let $H_B(R, P) := \{(s, y) : \sum_{i=1}^{k} \psi_B(r_i)s_i + \sum_{i=1}^{\ell} \widetilde{\psi_B}(p_i)y_i \geq 1\}$. Let $\mathcal{G}_b$ denote the set of all generalized cross-polyhedra (as applicable to $S = b + \mathbb{Z}^n$). Then, we have

THEOREM 1.6. *Let $n \in \mathbb{N}$ and $b \in \mathbb{Q}^n \setminus \mathbb{Z}^n$. Define for any matrices R, P*

$$M(R, P) := \cap_{B \text{ maximal } b+\mathbb{Z}^n \text{ free set}} H_B(R, P)$$
$$G(R, P) := \cap_{B\in\mathcal{G}_b} H_B(R, P)$$

Then there exists a constant α depending only on n, b such that $M(R, P) \subseteq G(R, P) \subseteq \alpha M(R, P)$ for all matrices R, P.

Note that since $\psi_B, \widetilde{\psi}_B \geq 0$, both $M(R, P)$ and $G(R, P)$ in Theorem 1.6 are polyhedra of the blocking type, i.e., they are contained in the nonnegative orthant and have their recession cone is the nonnegative orthant. Thus, the relationship $G(R, P) \subseteq \alpha M(R, P)$ shows that one can "blow up" the closure $M(R, P)$ by a factor of α and contain $G(R, P)$. Equivalently, if we optimize any linear function over $G(R, P)$, the value will be an α approximation compared to optimizing the same linear function over $M(R, P)$.

4. We test our family of cutting planes on randomly generated mixed-integer linear programs, on vertex cover and stable set problems in random graphs, and on the MIPLIB 3.0 set of problems. The short summary is that we seem to observe a tangible improvement with our cuts on the general random instances, *no improvement whatsoever* in the random graph instances, and no significant improvement on structured problems like MIPLIB 3.0 problems

(except for a specific family). The random data set consists of approx. 13000 instances, and our observed improvement cannot be explained by random noise. More details are available in section 4.

Our conclusion is that while the family of generalized cross polyhedra has a closure with good properties (like Theorem 1.6 above) and any particular cut from the family can be generated with light computational overhead (point 2. above), the *cut selection problem* is overwhelming even for this specialized family. We used a very naive random sampling method for selecting cuts from this family and clearly this heuristic is not good enough, as our computational results show. Our efforts at approximating the closure did not report anything different (see discussion in Section 4.4).

The one encouraging message we draw from our computational experience is that in the general random instances distinct gain was observed in a non-trivial fraction (about 10%). Perhaps this suggests that the cuts are able to exploit some structure in dense MIP problems. But what this structure could be is not very clear.

1.2. Discussion. We isolate a *parametrizable* family of $b + \mathbb{Z}^n$ free sets such that the cut generating functions derived from them are simultaneously "good" in the sense that their closure provides a good approximation to the closure of cuts obtained from all $b + \mathbb{Z}^n$ free sets, and "efficient" in the sense that the cut coefficients can be computed in a few lines of computer code. We are unaware of a similar result on cut generating functions from the literature (we do a more detailed literature review below).

While there are results in prior literature (discussed in the next subsection) that show the existence of "good" families in the sense of approximations, one potential concern with these families is the following. It seems impossible to give a "nice" parametrization of these families from [3] that can be exploited computationally. In contrast, the family we propose in this manuscript can be parametrized very cleanly by tuples of the form (γ, μ, U) where $\gamma \in \mathbb{R}^n, \mu \in \Delta^{n-1}$ (Δ^{n-1} is the standard simplex in $\mathbb{R}^n$) and $U \in \mathbb{R}^{n \times n}$ is a unimodular matrix (n refers to the chosen number of rows from the simplex tableaux on which the analysis is being done).

Moreover, the problem of actually computing the cut coefficients is highly non-trivial for these "good" families from the literature (involving closest lattice vector problems, as discussed in point 2. above). The only family of sets in previous literature where more efficient algorithms exist to compute *any* lifting is the family of 2-dimensional $b + \mathbb{Z}^n$ free convex sets and even there, it is ironically quite non-trivial to compute the trivial lifting [33]. But for the "good" family we propose above, even in arbitrary dimensions, we give an efficient algorithm to compute the trivial lifting (which also happens to be the unique minimal lifting).

We view the computation section as a proof-of-concept to illustrate that each step mentioned in the paper — constructing the generalized crosspolyhedra, computing their gauge and computing the trivial trival lifting — is constructive and hence implementable. That said, we have not been able to address the *cut selection problem* adequately in practice. Our family is still "too big" in spite of being "efficient" in the sense described above, and our heuristics for selecting cutting planes from this family were unable to provide the theoretical gains promised by the closure. We view the results in this paper as making some partial progress towards answering Question 1.4. There is no doubt that more advances are needed towards settling this question in a completely satisfying manner.

1.3. Related literature and discussion. It would be hard to list the numerous papers that have appeared in the last decade pertaining to cut generating functions. We refer to the reader to the recent surveys [26, 10, 15, 16] and [22, Chapter 6], and the references therein. There are some papers worth singling out as they relate more directly to the flavor of questions we investigate in this paper.

In [32, 33], the authors are explicitly concerned with computing the trivial lifting formula (1.3), without solving an integer linear program. In fact, our result outlined in Item 2 above is very much inspired by ideas from [33]. This, to the best of our knowledge, summarizes the most directly comparable literature on the *efficiency* aspect of cut generating functions. There also has been parallel work on the *goodness* aspect. The papers [7, 1, 12, 18, 23, 37, 25, 13, 24, 3] provide results that, from a rigorous mathematical perspective, either show that a certain subset of cut generating functions forms a good approximation, or some natural subset (like split cuts) forms a bad approximation in the worst case.

In general, testing of cut generating functions computationally, with and without the trivial lifting, has been done in [39, 42, 40, 8, 27, 31]. Perhaps the best summary of these investigations is a quote from Conforti, Cornuéjols and Zambelli [20]: "Overall, the jury is still out on the practical usefulness of [cut generating functions]" (the part in brackets is our paraphrasing of the original quote). Nevertheless, it is our firm belief that this only indicates further investigations with a computational perspective are needed in this area. We hope the results of this paper can guide this research. While our computational experience adds to the ambiguity of whether these new cutting plans are useful in practice, it is heartening (at least to us) to see the appreciable advantage observed in random instances. Moreover, some of the positive results reported in [31] came from using special cases of our construction of generalized cross-polyhedra.

1.4. Outline. The remainder of the paper is dedicated to rigorously establishing the above results. Section 2 formally introduces the class of generalized cross-polyhedra and Theorem 1.6 is proved. Section 3 then gives an algorithm for computing the trivial lifting for the family of generalized cross-polyhedra, which avoid solving integer linear programming problems or closest lattice vector problems for this purpose. Section 4 gives the details of our computational testing. Section 5 proves Theorem 1.5.

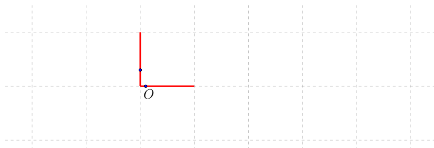
2. Approximation by Generalized Cross Polyhedra.

DEFINITION 2.1. *[Generalized cross-polyhedra] We define the family of* **generalized cross-polytopes** *recursively. For $n = 1$, a generalized cross-polytope is simply any interval $I_a := [a, a+1]$, where $a \in \mathbb{Z}$. For $n \geq 2$, we consider any generalized cross-polytope $B \subseteq \mathbb{R}^{n-1}$, a point $c \in B$, $\gamma \in \mathbb{R}$, and $\mu \in (0,1)$. A generalized cross-polytope in $\mathbb{R}^n$ built out of B, c, γ, μ is defined as the convex hull of $\left(\frac{1}{\mu}(B - c) + c\right) \times \{\gamma\}$ and $\{c\} \times \left(\frac{1}{1-\mu}(I_{\lfloor \gamma \rfloor} - \gamma) + \gamma\right)$. The point $(c, \gamma) \in \mathbb{R}^n$ is called the* **center** *of the generalized cross-polytope.*

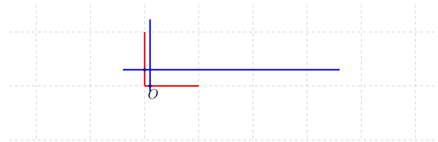
A **generalized cross-polyhedron** *is any set of the form $X \times \mathbb{R}^{n-m}$, where $m < n$ and $X \subseteq \mathbb{R}^m$ is a generalized cross-polytope in $\mathbb{R}^m$.*

The following theorem collects important facts about generalized cross-polyhedra that were established in [2, 17] (where these sets were first defined and studied) and will be important for us below.

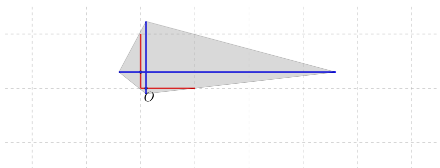
THEOREM 2.2. *Let $G \subseteq \mathbb{R}^n$ be a generalized cross-polyhedron. The following are all true.*



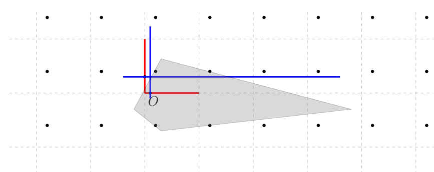
(a) The horizontal red line is the crosspolytope B and the vertical red line represents the interval $I_{\lfloor \gamma \rfloor}$. The points on B and $I_{\lfloor \gamma \rfloor}$ are c and γ respectively.



(b) With $\mu = 0.25$, the horizontal blue line is $\left(\frac{1}{\mu}(B - c) + c\right) \times \{\gamma\}$ and the vertical line is $\{c\} \times \left(\frac{1}{1-\mu}\left(I_{\lfloor \gamma \rfloor} - \gamma\right) + \gamma\right)$.



(c) The convex hull of the sets in Figure 2.1b gives G, the generalized cross-polytope.



(d) $b + G$ is the new $b + \mathbb{Z}^n$ free generalized cross-polytope.

Fig. 2.1: Cross-polytope construction - The points $b + \mathbb{Z}^n$ are shown as black dots and the points in $\mathbb{Z}^n$ are the intersection of the dotted grid.

(i) Let $b \in \mathbb{R}^n \setminus \mathbb{Z}^n$ such that $-b \in \operatorname{int}(G)$. Then $b + G$ is a maximal $b + \mathbb{Z}^n$ free convex set. Moreover, using the values of c, γ and μ in the recursive construction, one can find normal vectors $a^1, \ldots, a^{2^n} \in \mathbb{R}^n$ such that $b + G = \{x \in \mathbb{R}^n : a^i \cdot x \leq 1, \;\; i = 1, \ldots, 2^n\}$.

(ii) If G is a generalized cross-polytope, then there exists a unique $z \in \mathbb{Z}^n$ such that $z + [0,1]^n \subseteq G \subseteq \cup_{j=1}^{n}((z + [0,1]^n) + \ell_j)$, where ℓ_j is the line in $\mathbb{R}^n$ through the origin in the direction of the j-th unit vector. Moreover, $z_j = \lfloor \gamma_j \rfloor$, where γ_j is the value used in the j-th stage in the recursive construction of G for $j = 1, \ldots, n$ (for $j = 1$, γ_1 is taken to be the left end point of the interval used to start the construction).

Part (i) of Theorem 2.2 follows from [2, Theorem 5.3], or its generalization [17, Theorem 4.1]. Part (ii) follows from a straightforward inductive argument which we omit in this paper.

Next we recall the definition of lattice width and the flatness theorem, which we need to prove Theorem 1.6.

Definition 2.3 (Width function and lattice width). *For every nonempty subset $X \subset \mathbb{R}^n$, the* width function $w(X, \circ) : \mathbb{R}^n \mapsto [0, \infty]$ *of X is defined to be*

$$w(X, u) \quad := \quad \sup_{x \in X} x \cdot u - \inf_{x \in X} x.u \tag{2.1}$$

The lattice width *of X is defined as*

$$w(X) \quad := \quad \inf_{u \in \mathbb{Z}^n \setminus \{0\}} w(X, u) \tag{2.2}$$

Definition 2.4 (Flatness). *The* Flatness function *is defined as*

$$\mathrm{Flt}(n) \quad := \quad \sup\{w(B) : B \textit{ is a } b+\mathbb{Z}^n \textit{ free set in } \mathbb{R}^n\} \tag{2.3}$$

Theorem 2.5. *[6, Flatness theorem]* $\mathrm{Flt}(n) \le n^{5/2}$ *for all* $n \in \mathbb{N}$.

The main goal of this section is to establish the following result, which immediately implies Theorem 1.6.

Theorem 2.6. *Let $b \in \mathbb{Q}^n \setminus \mathbb{Z}^n$ such that the largest denominator in a coordinate of b is s. Let L be a $b+\mathbb{Z}^n$ free set with $0 \in \mathrm{int}(L)$. Then there exists a generalized cross-polyhedron G such that $B := b + G$ is a $b+\mathbb{Z}^n$ free convex set such that* $\left(\frac{1}{s4^{n-1}\,\mathrm{Flt}(n)}\right)^{n-1} L \subseteq B$.

Let us quickly sketch why Theorem 2.6 implies Theorem 1.6.

Proof of Theorem 1.6. We claim that $\alpha = \left(s4^{n-1}\,\mathrm{Flt}(n)\right)^{n-1}$ works. Gauge functions satisfy the properties that $A \subseteq B$ implies that $\psi_A \ge \psi_B$, and $\psi_{\gamma A} = \frac{1}{\gamma}\psi_A$ for any $\gamma \ge 0$ [43]. Thus, Theorem 2.6 implies that for any maximal $b+\mathbb{Z}^n$ free set L, there exists a generalized cross-polyhedron B such that $\psi_B \le \alpha\psi_L$, consequently, by (1.3), $\tilde{\psi}_B \le \alpha\tilde{\psi}_L$. Thus, $H_B(R,P) \subseteq \alpha H_L(R,P)$ and we are done. ∎

The rest of this section is dedicated to proving Theorem 2.6. We need to first introduce some concepts and intermediate results, and the final proof of Theorem 2.6 is assembled at the very end of the section.

Definition 2.7 (Truncated cones and pyramids). *Given an $n-1$-dimensional closed convex set $M \subset \mathbb{R}^n$, a vector $v \in \mathbb{R}^n$ such that $\mathrm{aff}(v+M) \ne \mathrm{aff}(M)$, and a scalar $\gamma \in \mathbb{R}_+$, we say that the set $T(M,v,\gamma) := \mathrm{cl}(\mathrm{conv}\{M \cup (\gamma M + v)\})$ is a* truncated cone *(any set that can be expressed in this form will be called a truncated cone).*

A truncated cone with $\gamma = 0$ is called a pyramid *and is denoted $P(M,v)$. If M is a polyhedron, then $P(M,v)$ is a* polyhedral pyramid*. v is called the apex of $P(M,v)$ and M is called the base of $P(M,v)$. The* height *of a pyramid $P(M,v)$ is the distance of v from the affine hull of M.*

When M is a hyperplane, the truncated cone is called a split*.*

Definition 2.8 (Simplex and Generalized Simplex). *A simplex is the convex hull of affinely independent points. Note that a simplex is also a pyramid. In fact, any facet of the simplex can be taken as the base, and the height of the simplex can be defined with respect to this base.*

A generalized simplex *in $\mathbb{R}^n$ is given by the Minkowski sum of a simplex Δ and a linear space X such that X and $\mathrm{aff}(\Delta)$ are orthogonal to each other. Any facet of $\Delta + X$ is given by the Minkowski sum of a base of Δ and X. The height of the generalized simplex with respect to such a facet is defined as the height of Δ with respect to the corresponding base.*

We first show that $b+\mathbb{Z}^n$ free generalized simplices are a good class of polyhedra to approximate other $b+\mathbb{Z}^n$ free convex bodies within a factor that depends only on the dimension. This result is a mild strengthening of Proposition 29 in [3] and the proof here is very similar to the proof of that proposition.

Lemma 2.9. *Let $n \in \mathbb{N}$ and $b \in \mathbb{Q}^n \setminus \mathbb{Z}^n$ such that the largest denominator in a coordinate of b is s. Let $S = b + \mathbb{Z}^n$. Then for any S-free set $L \subseteq \mathbb{R}^n$, there exists an S-free generalized simplex $B = \Delta + X$ (see Definition 2.8) such that* $\frac{1}{s4^{n-1}\,\mathrm{Flt}(n)} L \subseteq B$.

Moreover, after a unimodular transformation, B has a facet parallel to $\{x \in \mathbb{R}^n : x_n = 0\}$, the height of B with respect to this facet is at most 1, and $X = \mathbb{R}^m \times \{0\}$ for some $m < n$.

Proof. We proceed by induction on n. For $n = 1$, all S-free sets are contained in a $b + \mathbb{Z}$ free interval, so we can take B to be this interval. For $n \geq 2$, consider an arbitrary S-free set L. By Theorem 2.5, $L' := \frac{1}{s4^{n-2}\operatorname{Flt}(n)} L$ has lattice width at most $\frac{1}{s}$. Perform a unimodular transformation such that the lattice width is determined by the unit vector e^n and $b_n \in [0, 1)$.

If $b_n \neq 0$, then $b_n \in [1/s, 1 - 1/s]$, and therefore L' is contained in the split $\{x : b_n - 1 \leq x_n \leq b_n\}$. We are done because all splits are generalized simplices and $\frac{1}{s4^{n-1}\operatorname{Flt}(n)} L = \frac{1}{4} L' \subseteq L' \subseteq B := \{x : b_n - 1 \leq x_n \leq b_n\}$.

If $b_n = 0$, then $L \cap \{x : x_n = 0\}$ is an S'-free set in $\mathbb{R}^{n-1}$, where $S' = (b_1, \ldots, b_{n-1}) + \mathbb{Z}^{n-1}$. Moreover, by the induction hypothesis applied to $L \cap \{x : x_n = 0\}$ and $L' \cap \{x : x_n = 0\}$ it follows that there exists an S'-free generalized simplex $B' \subseteq \mathbb{R}^{n-1} \times \{0\}$ such that $L' \cap \{x : x_n = 0\} \subseteq B'$. Let B' be the intersection of halfspaces $H'_1, \ldots, H'_k \subseteq \mathbb{R}^{n-1}$. By a separation argument between L' and $\operatorname{cl}(\mathbb{R}^{n-1} \setminus H'_i) \times \{0\}$, one can find halfspaces $H_1, \ldots, H_k \subseteq \mathbb{R}^n$ such that $H_i \cap (\mathbb{R}^{n-1} \times 0) = H'_i \times \{0\}$ and $L' \subseteq H_1 \cap \ldots \cap H_k$ (this separation is possible because $0 \in \operatorname{int}(L')$).

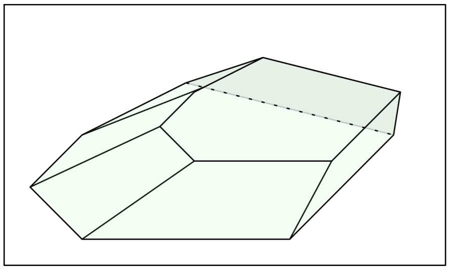
We now consider the set $P := H_1 \cap \ldots \cap H_k \cap \{x : -1/s \leq x_n \leq 1/s\}$. By construction, $P \subseteq \mathbb{R}^n$ is S-free and $L' \subseteq P$ since L' has height at most $\frac{1}{s}$ and contains the origin. P is also a truncated cone given by $v = \frac{2}{s} e^n$ and $M = P \cap \{x : x_n = -1/s\}$ and some factor γ (see Definition 2.7), because B' is a generalized simplex. Without loss of generality, one can assume $\gamma \leq 1$ (otherwise, we change v to $-v$ and M to $P \cap \{x : x_n = 1\}$). By applying Lemma 25 (b) in [3], one can obtain a generalized simplex B as the convex hull of some point $x \in P \cap \{x : x_n = \frac{1}{s}\}$ and M such that $\frac{1}{4} P \subseteq B \subseteq P$ (the hypothesis for Lemma 25 (b) in [3] is satisfied because 0 can be expressed as the mid point of two points in $P \cap \{x : x_n = \frac{1}{s}\}$ and $P \cap \{x : x_n = -\frac{1}{s}\}$). Since $L' \subseteq P$, we have that $\frac{1}{s4^{n-1}\operatorname{Flt}(n)} L = \frac{1}{4} L' \subseteq \frac{1}{4} P \subseteq B$. Since $B \subseteq P$, B is S-free. ∎

Proof of Theorem 2.6. We proceed by induction on n. If $n = 1$, then an S-free convex set is contained in an S-free interval, which is an S-free generalized cross-polyhedron, so we are done.

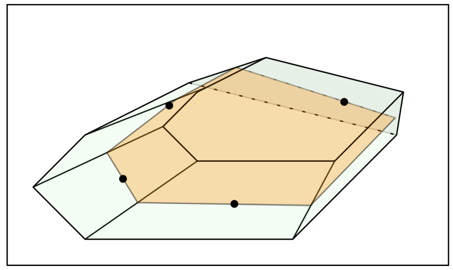
For $n \geq 2$, by Lemma 2.9, there exists an S-free generalized simplex $P = \Delta + X$ (see Definition 2.8) such that $\frac{1}{s4^{n-1}\operatorname{Flt}(n)} L \subseteq P$. Moreover, after a unimodular transformation, P has a facet parallel to $\{x \in \mathbb{R}^n : x_n = 0\}$ and the height of P with respect to this facet is at most 1. Moreover, X can be assumed to be $\mathbb{R}^m \times \{0\}$ for some $m < n$ since X has to be parallel to the facet defined by $x_n = 0$. Thus, by projecting on to the last $n - m$ coordinates, we may assume that P is a simplex with a facet parallel to $\{x \in \mathbb{R}^n : x_n = 0\}$. Without loss of generality, we may assume $b_n \in [0, 1)$ (by translating everything by an integer vector). We now consider two cases.

If $b_n \neq 0$, then $b_n \in [1/s, 1 - 1/s]$. Moreover, $\frac{1}{s} P$ has height at most $\frac{1}{s}$, and therefore it is contained in the maximal S-free split $\{x : b_n - 1 \leq x_n \leq b_n\}$. We are done because all maximal S-free splits are generalized cross-polyhedra and $\left(\frac{1}{s4^{n-1}\operatorname{Flt}(n)}\right)^{n-1} L \subseteq \frac{1}{s} P \subseteq B := \{x : b_n - 1 \leq x_n \leq b_n\}$.

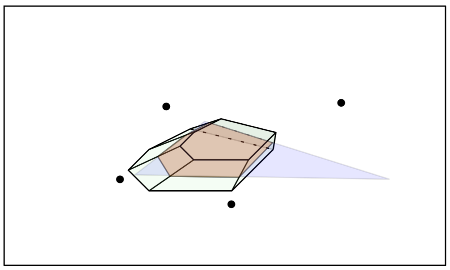
If $b_n = 0$, then by the induction hypothesis, there exists a translated generalized



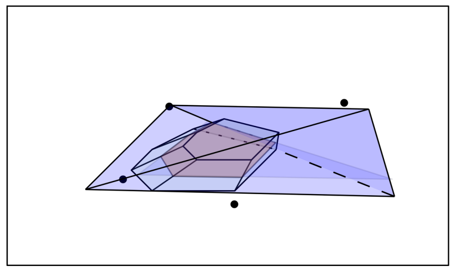
(a) A $b+\mathbb{Z}^n$-free convex set that is to be approximated with a $b+\mathbb{Z}^n$-free simplex.



(b) The integer lattice plane passing through the convex set is shown in orange.



(c) The set shown in orange is a *lower-dimensional* $b+\mathbb{Z}^n$-free convex set. This can be approximated by a lower-dimensional simplex using the induction hypothesis.



(d) Hyperplanes can be added that passes through the facets of the set in orange to get a truncated pyramid and then a simplex to approximate the given $b+\mathbb{Z}^n$-free set.

Fig. 2.2: Intuition behind Lemma 2.9 to approximate a $b+\mathbb{Z}^n$-free convex set with a simplex.

cross-polyhedron $B' \subseteq \mathbb{R}^{n-1}\times\{0\}$ such that $\left(\frac{1}{s4^{n-2}\,\mathrm{Flt}(n-1)}\right)^{n-2}(P\cap\{x : x_n = 0\}) \subseteq B'$. Let v be the vertex of P with positive v_n coordinate. Since the height of P is at most 1, the height of $\left(\frac{1}{s4^{n-2}\,\mathrm{Flt}(n-1)}\right)^{n-2}P$ is also at most 1. Let the facet F of $\left(\frac{1}{s4^{n-2}\,\mathrm{Flt}(n-1)}\right)^{n-2}P$ parallel to $\{x\in\mathbb{R}^n : x_n = 0\}$ be contained in the hyperplane $\{x\in\mathbb{R}^n : x_n = \lambda\}$, where $-1<\lambda<0$ since P has height at most 1 with respect to this facet. Moreover, we may assume that after a unimodular transformation, the projection of v on to $\mathbb{R}^{n-1}\times\{0\}$ lies in B', because the points from S on the boundary of B' form a lattice hypercube in $\mathbb{R}^{n-1}$ by Theorem 2.2(ii). Let this projected vertex be $c\in\mathbb{R}^{n-1}$. Let $\mu = 1-|\lambda|$ and $\gamma=\lambda$. Create the generalized cross-polyhedron B from B', c, μ, γ in $\mathbb{R}^n$ as described in Definition 2.1. By the choice of μ and γ and the fact that P has height at most 1, $v\in B$.

We also claim that $F \subseteq (\frac{1}{\mu}(B'-c)+c)\times\{\gamma\}\subseteq B$. Indeed, observe that

$$F-(c,\lambda) \subseteq \frac{1}{\mu}\left(\left(\left(\frac{1}{s4^{n-2}\,\mathrm{Flt}(n-1)}\right)^{n-2}P\cap\{x\in\mathbb{R}^n : x_n=0\}\right)-(c,0)\right).$$

Since $\left(\frac{1}{s4^{n-2}\,\mathrm{Flt}(n-1)}\right)^{n-2}(P \cap \{x : x_n = 0\}) \subseteq B'$, we have $F \subseteq \left(\frac{1}{\mu}(B'-c)+c\right) \times \{\gamma\}$.

Thus, we have that $\left(\frac{1}{s4^{n-2}\,\mathrm{Flt}(n-1)}\right)^{n-2} P \subseteq B$ since $v \in B$ and $F \subseteq B$. Combining with $\frac{1}{s4^{n-1}\,\mathrm{Flt}(n)} L \subseteq P$, we obtain that

$$\left(\frac{1}{s4^{n-1}\,\mathrm{Flt}(n)}\right)^{n-1} L \subseteq \left(\left(\frac{1}{s4^{n-2}\,\mathrm{Flt}(n-1)}\right)^{n-2}\right) \frac{1}{s4^{n-1}\,\mathrm{Flt}(n)} L \subseteq B$$

3. Algorithms for trivial lifting in generalized cross-polyhedra. The key fact that we utilize in designing an algorithm to compute the trivial liftings of generalized cross-polyhedra is the following: generalized cross-polytopes have the so-called *covering property*. We refer the readers to [9] for the implications that the covering property leads to and especially to [9, Theorem 5] which shows that the covering property is necessary and sufficient to ensure that the trivial lifting is the unique minimal lifting.

[17, Section 4] discusses the coproduct operation used to construct the generalized cross-polytopes. [17, Theorem 4.1] assures that as long as the "initial" sets used in the coproduct operation have the covering property, so does the final set. In our construction of generalized cross-polytopes, the corresponding initial sets are $b+\mathbb{Z}$ free intervals, which have the covering property.

Having the covering property is important for computations in the following way: it implies existence of the so-called lifting region T (first defined in [29]) corresponding to the generalized cross-polyhedra such that $T + \mathbb{Z}^n = \mathbb{R}^n$. Then, one can calculate the trivial lifting at a point x by calculating the gauge at $x+z$ where $z \in \mathbb{Z}^n$ and $x + z \in T$ (such a z always exists because $T + \mathbb{Z}^n = \mathbb{R}^n$). We formalize this in the theorem below.

Theorem 3.1. *Let $G \subseteq \mathbb{R}^m$ be any generalized cross-polytope and let $b \in \mathbb{R}^m \setminus \mathbb{Z}^m$ such that $-b \in \mathrm{int}(G)$. There is a subset $T \subseteq G$ such that $T + \mathbb{Z}^m = \mathbb{R}^m$ and for any $p \in \mathbb{R}^m$, there exists $\tilde{p} \in b + T$ such that $\tilde{p} \in p + \mathbb{Z}^m$ and $\widetilde{\psi_{b+G}}(p) = \psi_{b+G}(\tilde{p})$.*

Thus, for any generalized cross-polyhedron $G \subseteq \mathbb{R}^m$ and $p \in \mathbb{R}^m$, if one can find the $\tilde{p}$ in Theorem 3.1, then one can compute the trivial lifting coefficient $\tilde{\psi}_{b+G}(p)$ by simply computing the gauge function value $\psi_{b+G}(\tilde{p})$. The gauge function can be computed by simple evaluating the 2^m inner products in the formula $\psi_{b+G}(r) = \max_{i=1}^{2^m} a^i \cdot r$, where $a^i, i = 1, \ldots, 2^m$ are the normal vectors as per Theorem 2.2(i).

Thus, the problem boils down to finding $\tilde{p}$ from Theorem 3.1, for any $p \in \mathbb{R}^m$. Here, one uses property (ii) in Theorem 2.2. This property guarantees that given a generalized cross-polytope $G \subseteq \mathbb{R}^m$, there exists $\bar{z} \in \mathbb{Z}^n$ that can be explicitly computed using the γ values used in the recursive construction, such that $T \subseteq G \subseteq \cup_{j=1}^m((\bar{z} + [0,1]^m) + \ell_j)$, where ℓ_j is the 1-dimensional linear subspace parallel to the j-th coordinate axis obtained by setting all coordinates to 0 except coordinate j. Now, for any $p \in \mathbb{R}^m$, one can first find the (unique) translate $\hat{p} \in p + \mathbb{Z}^n$ such that $\hat{p} \in b + \bar{z} + [0,1]^m$ (this can be done since b and z are explicitly known), and then $\tilde{p}$ in Theorem 3.1 must be of the form $\hat{p} + Me^j$, where $M \in \mathbb{Z}$ and e^j is one of the standard unit vectors in $\mathbb{R}^m$. Thus,

$$\tilde{\psi}_{b+G}(p) = \min_{\substack{j \in \{1,\ldots,m\},\\ M \in \mathbb{Z}}} \psi_{b+G}(\hat{p} + Me^j).$$

For a fixed $j \in \{1, \ldots, m\}$, this is a one dimensional convex minimization problem over the integers $M \in \mathbb{Z}$ for the piecewise linear convex function $\phi_j(\lambda) = \psi_{b+G}(\hat{p} + \lambda e^j) = \max_{i=1}^{2^m} a^i \cdot (\hat{p} + \lambda e^j)$. Such a problem can be solved by simply sorting the slopes of the piecewise linear function (which are simply $a^1_j, \ldots, a^{2^n}_j$), and finding the point $\bar{\lambda}$ where the slope changes sign. Then either $\phi_j(\lceil \bar{\lambda} \rceil)$ or $\phi_i(\lfloor \bar{\lambda} \rfloor)$ minimizes ϕ_j. Taking the minimum over $j = 1, \ldots, m$ gives us the trivial lifting value for p.

One observes that this entire procedure takes $O(m2^m)$. While this was described only for generalized cross-polytopes, generalized cross-polyhedra of the form $G \times \mathbb{R}^{n-m}$ pose no additional issues: one simply projects out the $n - m$ extra dimensions.

We give a formal description of the algorithm below in Algorithm 3.1. We assume access to procedures GetNormal(G, b) and Gauge(G, b, x). GetNormal(G, b) takes as input a generalized cross-polytope G and b such that $-b \in \text{int}(G)$, and returns the list of normals $\{a^1, \ldots, a^{2^n}\}$ such that $b + G = \{x \in \mathbb{R}^n : a^i \cdot x \leq 1, \;\; i = 1, \ldots, 2^n\}$ (property (i) in Theorem 2.2). Gauge(G, b, r) takes as input a generalized cross-polytope G and b such that $-b \in \text{int}(G)$ and a vector r, and returns $\psi_{b+G}(r)$ (given the normals from GetNormal(G, b), one simply computes the 2^n inner products $a^i \cdot r$ and returns the maximum).

Algorithm 3.1 Trivial lifting of a generalized cross-polytope

Input: Generalized cross-polytope $G \subseteq \mathbb{R}^n$, $b \in \mathbb{R}^n \setminus \mathbb{Z}^n$ such that $-b \in \text{int}(G)$. $p \in \mathbb{R}^n$ where the lifting is to be evaluated.

Output: $\widetilde{\psi_{b+G}}(p)$

function CrossPolyLift(G, $\mathbf{b}$, $\mathbf{x}$)
    Set $\bar{z} \in \mathbb{R}^n$ using parameters of G as given in property (ii) in Theorem 2.2.
    Compute unique $\hat{p} \in (p + \mathbb{Z}^n) \cap \mathbf{b} + \bar{z} + [0, 1]^n$.
    Let $\mathcal{N} =$ GetNormal(G, b) be the set of normals.
    for Each coordinate j from 1 to n **do**
        Find $a^- \in \arg\max_{a \in \mathcal{N}} \{a_j : a_j \leq 0\}$ where a_j denotes the j-th coordinate of $a \in \mathcal{N}$). Break ties by picking the one with maximum $a \cdot \hat{p}$.
        Find $a^+ \in \arg\min_{a \in \mathcal{N}} \{a_j : a_j > 0\}$ where a_j denotes the j-th coordinate of $a \in \mathcal{N}$). Break ties by picking the one with maximum $a \cdot \hat{p}$.
        $\bar{\lambda} \leftarrow \frac{a^+ \cdot \hat{p} - a^- \cdot \hat{p}}{a^- - a^+}$.
        $m_j \leftarrow \min\{a^+ \cdot \hat{p} + \lceil \bar{\lambda} \rceil a^+_j, a^- \cdot \hat{p} + \lfloor \bar{\lambda} \rfloor a^-_j\}$.
    end for
    return $\min\{1, m_1, \ldots, m_j\}$.
end function

4. Computational Experiments and Results. In this section we give results from a set of computational experiments comparing the cuts described in this paper against Gomory's Mixed Integer (GMI) cuts, and also CPLEX computations at the root node. We perform four types of computational tests:

1. Testing on random dense instances of pure-integer and mixed-integer programs.
2. Testing stable-set problem instances and vertex-cover problem instances in random graphs.
3. Testing on MIPLIB3.0 problem instances.
4. Testing an approximation to the closure of the cuts from all generalized cross-polytopes, on the random dense instances of mixed-integer programs.

In the following subsections, we describe the terms used above, the exact testing procedure adopted and our results in these problems. We observe that, despite the strong theoretical results, the performance of the cuts derived from the generalized cross-polytopes in our particular computational set-ups is generally poor. As mentioned in the Introduction, we suspect that this is because our naive sampling of the cuts is not good enough and the cut selection problem for this family we propose is still a non trivial problem.

4.1. Test on random dense instances. First we describe the test we performed on random dense instances of pure and mixed-integer programs. We describe our problem generation procedure, cut generating procedure, comparison procedure in the following paragraphs. The testing procedure is also summarized in Algorithm 4.1.

4.1.1. Data generation. e write all our test problems in the canonical form

$$\min_{x \in \mathbb{R}^d} \left\{ c^T x : Ax = b; x \geq 0; i \in \mathcal{I} \implies x_i \in \mathbb{Z} \right\} \tag{4.1}$$

where $A \in \mathbb{R}^{k \times d}, b \in \mathbb{R}^k, c \in \mathbb{R}^d$ and $\mathcal{I} \subseteq \{1, 2, \ldots, n\}$.

We generated roughly 12,000 problems in the following fashion.

- Each problem can be pure integer or mixed integer. For mixed-integer problem, we decide if each variable is discrete or continuous randomly with equal probability.
- Each problem can have the data for A, b and c as matrices with either integer data or rational data. Each entry is uniformly distributed between -10 and 10. In the former case, only integers are considered and in the latter case, rational numbers represented upto 8 decimal places are considered. Thus the matrix A is a dense matrix.
- The size of each problem varies from $(k, d) \in \{(10i, 25i) : i \in \{1, 2, \ldots, 10\}\}$.
- There are roughly 300 realizations of each type of problem.

This leads to $2 \times 2 \times 10 \times 300\,(\text{roughly}) \approx 12,000$ problems in all. The entire data set can be found at this hyperlink: http://www.ams.jhu.edu/~abasu9/Data_Sets/. This number is not precise as some random problems where infeasibility or unboundedness were discovered in the LP relaxation were ignored. Below we present the results for these approximately 12,000 problems as a whole and also the performance of our methods in various subsets of these instances.

4.1.2. Cut generation. We consider three types of cuts in these computational tests - Gomory's mixed-integer (GMI) cuts, X-cuts and GX-cuts. GMI cuts are single row cuts obtained from standard splits [22, Eqn 5.31]. GX-cuts are cuts obtained from certain structured generalized cross-polytopes defined in Definition 2.1. X-cuts are obtained from a special case of generalized cross-polytopes, where the center (c, γ) coincides with the origin. It should be noted that the GMIs are indeed a special case of X-cuts, because they can be viewed as cuts obtained from $b + \mathbb{Z}^n$ free intervals or one-dimensional generalized cross-polytopes whose center coincide with the origin. In this section, we call such cross-polytopes as *regular cross-polytopes*. This motivates the set inclusions shown in Figure 4.1. The motivation behind classifying a special family of cross-polytopes with centers coinciding with the origin is the algorithmic efficiency they provide. Because of the special structure in these polytopes, the gauges and hence the cuts can be computed much faster than what we can do for an arbitrary generalized cross-polytope (comparing with the algorithms in section 3). In particular, the gauge and the trivial lifting can both be computed in $O(n)$ time, as opposed to $O(2^n)$ and $O(n2^n)$ respectively for the general case (see section 3), where n is the

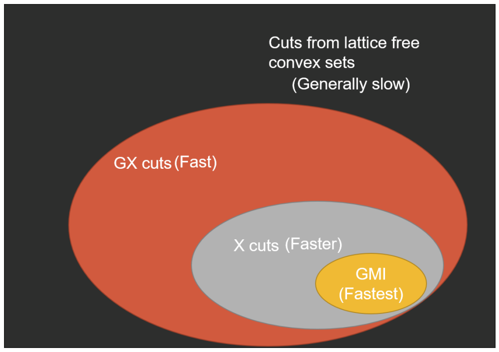


Fig. 4.1: Venn diagram showing inclusions of various types of cuts and algorithmic efficiencies to generate them.

dimension of the generalized cross-polytopes or equivalently, the number of rows of the simplex tableaux used to generate the cut.

The family of generalized cross-polytopes that we consider can be parameterized by a vector $\mu \in (0,1)^n$ and another vector in $f \in \mathbb{R}^n$. This vector consists of the values μ_i used in each stage of construction of the cross-polytope, after appropriate normalization (see Definition 2.1). This actually forces $\sum_{i=1}^n \mu_i = 1$. The vector f corresponds to the center of the generalized cross-polytope; the coordinates of f give the coordinates of c and γ in the iterated construction of Definition 2.1. Both the parameters μ and f show up in Algorithm 4.1. The regular cross-polytopes are obtained by setting $f = \mathbf{0}$ in the above construction; thus, they are parameterized by only the vector $\mu \in (0,1)^n$. As long as $\sum_{i=1}^n \mu_i = 1$, there exists a one-to-one map between such vectors and the set of regular cross-polytopes in $\mathbb{R}^n$.

We also note that *any* cut generated from a generalized cross-polytope, or for that matter, any valid pair (see Definition 1.1) cuts off the fractional solution. This is because, the fractional solution obtained corresponds to $s = 0$ and $y = 0$ in the space (s, y) using the notation in (1.2). So no matter what the values of $\psi(r_i)$ and $\pi(p_i)$ are, the LHS of the inequality in (1.2) is 0 for the fractional point. Thus the current fractional LP solution is *always* separated from the convex hull.

4.1.3. Comparison procedure. In each of the problems, the benchmark for comparison was an aggressive addition of GMI cuts. The procedure used for comparison is mentioned in Algorithm 4.1. We would like to emphasize that X-cuts and GX-cuts are an infinite family of cuts unlike the GMI cuts. However, we add only finitely many cuts from this infinite family.

In all the computational tests in this paper, these cuts are randomly generated without looking into any systematic selection of rows or μ. However to improve the performance from a completely random selection, we generate ℓ batches of k cuts and only keep the best set of k cuts. We lay out our testing procedure in detail in Algorithm 4.1.

For the set of 12,000 problems, X-cuts and GX-cuts were generated with $N =$ 2, 5, and 10 rows. For GX-cuts, the number q of rows to be picked whose corresponding basic variables violate integrality constraints, was chosen to be 1. This was found to be an ideal choice under some basic computational tests with small sample size, where cuts with different values of q were compared. Also, a qualitative motivation behind

Algorithm 4.1 Computational testing procedure

Input: A mixed-integer problem (MIP) in standard form. Number $N \geq 2$ of rows to use to generate multi-row cuts; Number $k \geq 1$ of multi-row cuts; Number $\ell \geq 1$ of rounds of multi-row cuts to be used; Number of $1 \leq q \leq N$ non-integer basics to be picked for GX-cuts.

$\mathsf{LP} \leftarrow$ Objective of LP relaxation of MIP.
In the final simplex tableaux, apply GMI cuts on all rows whose corresponding basic variables are constrained to be integer in the original problem, but did not turn out to be integers.
$\mathsf{GMI} \leftarrow$ Objective of LP relaxation of MIP and GMI cuts.
for i from 1 to ℓ **do**
    for j from 1 to k **do**
        Generate $\mu \in [0, 1]^N$ such that $\sum_{\xi=1}^{N} \mu_\xi = 1$. Also randomly select N rows where integrality constraints are violated for corresponding basic variables.
        Generate an X-cut from the generated μ and the chosen set of rows.
        Generate $f \in [0, 1]^N$ randomly.
        Randomly select rows such that q of them correspond to rows that violate the integrality contraints and $N - q$ of them don't.
        Generate a GX-cut from the generated μ, f and the set of rows.
    end for
    $\mathsf{X}_i \leftarrow$ Objective of LP relaxation of MIP and all the X-cuts generated above.
    $\mathsf{XG}_i \leftarrow$ Objective of LP relaxation of MIP with all the X-cuts as well as the GMI cuts.
    $\mathsf{GX}_i \leftarrow$ Objective of LP relaxation of MIP and all the GX-cuts generated above.
    $\mathsf{GXG}_i \leftarrow$ Objective of LP relaxation of MIP with all the GX-cuts as well as the GMI cuts.
end for
$\mathsf{X} \leftarrow \max_{i=1}^{\ell} \mathsf{X}_i$; $\mathsf{XG} \leftarrow \max_{i=1}^{\ell} \mathsf{XG}_i$; $\mathsf{GX} \leftarrow \max_{i=1}^{\ell} \mathsf{GX}_i$; $\mathsf{GXG} \leftarrow \max_{i=1}^{\ell} \mathsf{GXG}_i$.
$\mathsf{Best} \leftarrow \max \{\mathsf{X}, \mathsf{XG}, \mathsf{GX}, \mathsf{GXG}\}$
return LP, GMI, X, XG, GX, $\mathsf{GXG}, \mathsf{Best}$

choosing $q = 1$ is as follows: GMI cuts use information only from those rows where integrality constraints on the corresponding basic variables are violated. To beat GMI, it is conceivably more useful to use information not already available for GMI cuts, and hence to look at rows where the integrality constraint on the corresponding basic variable is not violated.

4.1.4. Results. A typical measure used to compute the performance of cuts is *gap closed* which is given by $\frac{\mathsf{cut}-\mathsf{LP}}{\mathsf{IP}-\mathsf{LP}}$. However the IP optimal value IP could be expensive to compute on our instances. So, as a first test, we use a different metric, which compares the performance of the best cut we have, against that of GMI cuts. Thus we define

$$\beta \quad = \quad \frac{\mathsf{Best} - \mathsf{GMI}}{\mathsf{GMI} - \mathsf{LP}}, \tag{4.2}$$

which tries to measure the *improvement* over GMI cuts using the new cuts.

The testing procedure mentioned in Algorithm 4.1 was run with the values of $k = \ell = 5$. The results hence obtained are mentioned in Table 4.1. Besides this table, we present some interesting observations from our computational testing.

Table 4.1: Results

Filter	**Number of problems**	**Cases where GMI < Best**	**Average of β**	**Average of β when GMI is beaten**
None (All problems)	13604	6538 (48.06%)	2.00%	4.15%
Rational Data	6600	3213 (48.68%)	2.11%	4.23%
Integer Data	7004	3325 (47.47%)	1.90%	3.80%
Pure Integer problems	6802	2189 (32.18%)	0.69%	2.146%
Mixed Integer problems	6802	4376 (64.33%)	3.32%	5.159%
Rational Data Pure Integer problems	3300	1078 (32.67%)	0.75%	2.306%
Rational Data Mixed Integer problems	3300	2135 (64.70%)	3.48%	5.376%
Integer Data Pure Integer problems	3502	1111 (31.52%)	0.63%	1.996%
Integer Data Mixed Integer problems	3502	2241 (63.42%)	3.17%	4.95%

1. In mixed-integer problems, we have $\beta \geq 10\%$ in 648 cases (which is 9.53% of the set of mixed-integer problems). In pure-integer problems we have $\beta \geq 5\%$ in 320 cases (which is 4.7% of the set of pure-integer problems). A conclusion from this could be that the family of cuts we are suggesting in this paper works best when we have a good mix of integer and continuous variables. We would like to remind the reader that in the mixed-integer examples we considered, roughly half the variables were continuous, due to a random choice between presence or absence of integrality constraint for each variable.
2. We also did some comparisons between $N = 2, 5, 10$ row cuts. In particular, let us define β_2, β_5 and β_{10} as the values of β with $N = 2, 5, 10$ respectively. Among the 13,604 cases, only in 265 cases we found $\beta_5 > \beta_2$ or $\beta_{10} > \beta_2$ (the inequalities are considered strictly here). In 264 of these cases, $\max\{\beta_5, \beta_{10}\} > \mathsf{GMI}$ (the inequality is strict here). In these 265 cases, 62 were pure-integer problems and GMI was beaten in all 62 problems. The other 203 cases were mixed integer problems. GMI was beaten in 202 of these problems.
   We conclude that when cuts derived from higher dimensional cross-polytopes dominate cut obtained from lower dimensional cross-polytopes, then the cuts from the higher dimensional cross-polytopes dominate GMI cuts as well. In other words, if we find a good cut from a high dimensional cross-polytope, then we have a very useful cut in the sense that it adds significant value over

GMI cuts.

3. Another test was done with increasing the number k which corresponds to the number of GX cuts added, from a constant 10 to half the number of GMI cuts in the problem (recall that for the results reported in Table 4.1, $k = 5$). Integer data was used in this, and this test was performed in a smaller randomly generated sample of size 810. In pure integer cases, we beat GMI in about 25% cases and in mixed-integer problems, we beat GMI in 61% cases. The value of β is comparable to Table 4.1 in both cases. But the lack of significant improvement suggests the following. The performance of cross-polytope based cuts is determined more by the problem instance characteristics, rather than the choice of cuts. If these cuts work well for a problem, then it should be reasonably easy to find a good cut.
4. Further there were 4 problems, all mixed-integer, with $\beta > 100\%$ suggesting potential that there could be a set of problems on whom a very good choice of rows and μ could give a non-trivial improvement over the GMI cuts.
5. As far as the time taken to run these instances goes, for the number of rows considered in this test, most of the time is typically spent in solving the LP relaxation after addition of cuts, accessing the simplex tableaux to generate the cut etc., rather than actually computing the cut.

4.2. Performance in random graph instances. Inspired by the notion that most of the integer programming problems of interest are sparse and have an underlying structure in them, we tested the cuts from the family of generalized cross-polyhedra on two graph problems namely, the stable set problem and the vertex cover problem. Both these problems are NP-complete by themselves and can be posed as an IP. Let $G = (V, E)$ be a graph. Equation (4.3) is the stable set problem and (4.4) is the vertex cover problem.

$$\text{(4.3a)} \qquad \max_{x_v} \; : \; \sum_{v \in V} x_v \qquad \text{subject to}$$

$$\text{(4.3b)} \qquad x_u + x_v \leq 1 \qquad \forall\, e = uv \in E$$

$$\text{(4.3c)} \qquad x_v \in \{0, 1\} \qquad \forall\, v \in V$$

$$\text{(4.4a)} \qquad \min_{x_v} \; : \; \sum_{v \in V} x_v \qquad \text{subject to}$$

$$\text{(4.4b)} \qquad x_u + x_v \geq 1 \qquad \forall\, e = uv \in E$$

$$\text{(4.4c)} \qquad x_v \in \{0, 1\} \qquad \forall\, v \in V$$

We generated the graphs as follows. We fixed the number of vertices $|V|$ and generated an edge e with a probability p. We generate 100 such instances for each value of $|V|$ and p. We varied $|V|$ from $5, 6, \ldots, 15$ and p from 0.1 to 0.9 in increments of 0.1. Both the stable set problem in (4.3) and the vertex cover problem in (4.4) problem were solved for these graphs.

We adopted a testing procedure analogous to the procedure mentioned in subsection 4.1. However, in this setting, we never observed *any* improvement whatsoever beyond the gain obtained using GMI cuts.

4.3. Performance in MIPLIB 3.0. Our testing with the new cuts discussed in this paper had meagre to no improvement in most of MIPLIB problems. Apart from the type of test mentioned in Algorithm 4.1 above, we performed the following test motivated by [31]. We ran the MIPLIB problem on CPLEX 12.7.1, stopping after all root node calculations before any branching begins (CPLEX typically adds several rounds of cuts at the root node itself). We keep count of number of cuts added by CPLEX. Now we allow up to 10 times the number of cuts added by CPLEX, iteratively solving the LP relaxation after the addition of each cut. In each round, the cut that gives the best β among twenty five randomly generated cut is added. We count the number of cuts we had to add and hence the number of rounds of LP we solve, to obtain an objective value as good as CPLEX. However, in almost all cases adding even ten times as many cuts as CPLEX did, did not give us the objective value improvement given by CPLEX.

Tests along the line of Algorithm 4.1 were also not promising. The only set of exceptions is the `enlight` set of problems in MIPLIB 3.0. These are problems coming from the Enlight combinatorial game. The X-cuts did not show any improvement over GMI cuts. The performance of the GX-cuts are shown below in Table 4.2. It can be seen from Table 4.2 that the performance of GX cuts increases with the number of rows used.

We note that we want to test the efficacy of our general purpose cutting planes, and therefore avoid using any knowledge of the structure of the MIPLIB problems in our cut generation procedure. While there could certainly be a way to use problem structure in deploying these cuts better in practice, we consider this more sophisticated approach to be outside the scope of this current manuscript.

Table 4.2: Performance on Enlight problems. The numbers reported are the optimal values of the LP after the corresponding cuts have been added (they are minimization problems).

Problem	**LP**	**GMI**	**2 row GX**	**5 row GX**	**10 row GX**	**IP**
enlight9	0	1	1.1902	1.4501	1.9810	INF
enlight13	0	1	1.1815	1.5410	1.9704	71
enlight14	0	1	1.1877	1.5051	1.9195	INF
enlight15	0	1	1.2001	1.4712	1.8991	69
enlight16	0	1	1.1931	1.4934	1.8766	INF

4.4. Approximating the exact closure of generalized cross polyhedra cutting planes. Using the β metric defined above, we see that the most significant improvement is on dense random instances. Thus, we tried to do a little more intensive testing on random dense instances by apronximating the exact closure of our family as best as we could. In other words, this is an attempt to optimize the linear function over the closure of the family of cuts obtained from generalized cross-polyhedra. Because of the nonlinear relation between the cut coefficients and the parameter μ used in defining the cross-polytope, implementing an exact separation oracle to solve this problem requires us to solve a nonlinear optimization problem. Moreover, the bigger hurdle seems to be the lack of any easy way to decide which rows should be selected to generate the separating cut from the family. This makes the separation problem for the exact closure a large mixed-integer nonlinear optimization problem which we did not see an efficient way to solve. To simulate the effect of the exact closure, we instead add a large number (∼1000) of random cuts from this family and compute the gap closed.

Since we are adding a lot of cuts compared to GMI, it makes more sense to consider the overall gap closed with respect to the optimal IP solution, as opposed to using the β metric. For large random dense instances, solving the IP to optimality is usually very difficult. So we decided to focus on set of about 200 random instances with 40 constraints and 100 variables.

In these 200 problems, the gap closed given by $\frac{\mathsf{Best-LP}}{\mathsf{IP-LP}}$ is of the order of 5.51%. In comparison, GMI cuts already close 5.04% of the gap. While this improvement is not very large, it is non trivial, in our opinion. It seems to complements the 10% improvement we saw in 10% of the cases when evaluating using the β metric (see point 1. in the discussion in Section 4.1.4). With the approximate closure this 10% improvement (going from ∼5% to ∼5.5%) is now seen to be an average phenomenon as opposed to only in 10% of the cases. Of course, one has to keep in mind that the approximate closure of our family uses a lot more cuts than the GMI closure; on the other hand, we are looking at gap closed as opposed to the β metric now, so these numbers still tell us something about our family.

5. Limitation of the trivial lifting: Proof of Theorem 1.5. In this section, we show that for a general $b + \mathbb{Z}^n$ free set, the trivial lifting can be arbitrarily bad compared to a minimal lifting. We first show that for $n = 2$, there exist $b \in \mathbb{R}^2 \setminus \mathbb{Z}^2$ such that one can construct maximal $(b+\mathbb{Z}^2)$-free triangles with the desired property showing that the trivial lifting of its gauge function can be arbitrarily worse than a minimal lifting.

Example in 2 dimensions:. Consider the sequence of Type 3 Maximal $b+\mathbb{Z}^n$ free triangles with $b = (-0.5, -0.5)$ given by the equations

$$20x - y + 10.5 = 0 \tag{5.1a}$$

$$\alpha_i x + y + \frac{1-\alpha_i}{2} = 0 \tag{5.1b}$$

$$-\beta_i x + y + \frac{1+\beta_i}{2} = 0 \tag{5.1c}$$

with $\alpha_i = 1 + \frac{1}{i}$ and $\beta_i = \frac{1}{i}$. Let us call the sequence of triangles as T_i. The triangle T_1 is shown in Fig. 5.1.

For all i, the point $p = (0.25, 0)$ is located outside the region $T_i + \mathbb{Z}^n$. So clearly for all i, the trivial lifting evaluated at p is at least 1. However, let us consider the minimum possible value any lifting could take at p. This is given by (see [29, Section 7], [14]):

$$\pi_{\min}(p) = \sup_{\substack{z\in\mathbb{Z}^n\\ w\in\mathbb{R}^n\\ w+Np\in b+\mathbb{Z}^n}} \frac{1-\psi_{T_i}(w)}{N} \tag{5.2}$$

$$= \sup_{\substack{N\in\mathbb{N}\\ z\in\mathbb{Z}^n}} \frac{1-\psi_{T_i}(b-Np+z)}{N} \tag{5.3}$$

$$= \sup_{N\in\mathbb{N}} \frac{1-\inf_{z\in\mathbb{Z}^n}\psi_{T_i}(b-Np+z)}{N} \tag{5.4}$$

$$= \sup_{N\in\mathbb{N}} \frac{1-\widetilde{\psi_{T_i}}(b-Np)}{N} \tag{5.5}$$

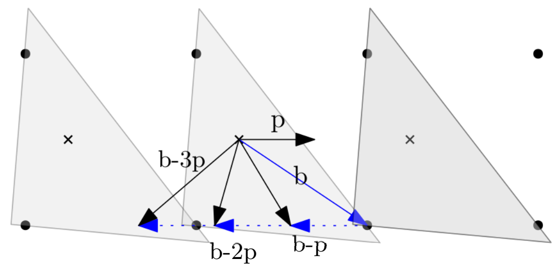


Fig. 5.1: Example where trivial lifting can be very poor

In the current example, $b = (-0.5, -0.5)$ and $p = (0.5, 0)$. Hence points of the form $b - Np$ correspond to a horizontal one-dimensional lattice. i.e., points of the form $(-(N+1)/2, -0.5)$. Since all of these points are arbitrarily close to the side of $T_i + z$ for some $z \in \mathbb{Z}^2$ (as $i \to \infty$), $\widetilde{\psi_{T_i}}(b - Np) \geq 1 - \varepsilon_i$ where $\varepsilon_i \to 0$. This implies that the minimal lifting of the point could become arbitrarily close to zero, and the approximation $\frac{\widetilde{\psi}(p)}{\pi_{\min}(p)}$ could be arbitrarily poor.

The proof for general $n \geq 2$ can be completed in two ways. One is a somewhat trivial way, by considering cylinders over the triangles considered above. A more involved construction considers the so-called *co-product* construction defined in [2, 17], where one starts with the triangles defined above and iteratively takes a co-product with intervals to get maximal $b + \mathbb{Z}^n$ free sets in higher dimensions. It is not very hard to verify that the new sets continue to have minimal liftings which are arbitrarily better than the trivial lifting, because they contain a lower dimension copy of the triangle defined above. We do not provide more details, because this will involve definitions of the coproduct construction and other calculations which do not provide any additional insight, in our opinion.

REFERENCES

[1] K. Andersen, C. Wagner, and R. Weismantel, *On an analysis of the strength of mixed-integer cutting planes from multiple simplex tableau rows*, SIAM Journal on Optimization, 20 (2009), pp. 967–982.

[2] G. Averkov and A. Basu, *Lifting properties of maximal lattice-free polyhedra*, Mathematical Programming, 154 (2015), pp. 81–111.

[3] G. Averkov, A. Basu, and J. Paat, *Approximation of corner polyhedra with families of intersection cuts*, in International Conference on Integer Programming and Combinatorial Optimization, Springer, Cham, 2017, pp. 51–62.

[4] E. Balas, *Intersection cuts – a new type of cutting planes for integer programming*, Operations Research, 19 (1971), pp. 19–39.

[5] E. Balas and R. G. Jeroslow, *Strengthening cuts for mixed integer programs*, European Journal of Operational Research, 4 (1980), pp. 224–234, https://doi.org/DOI:10.1016/0377-2217(80)90106-X, http://www.sciencedirect.com/science/article/pii/037722178090106X.

[6] A. Barvinok, *A Course in Convexity*, Providence: American Mathematical Society, 2002, https://doi.org/dx.doi.org/10.1090/gsm/054.

[7] A. Basu, P. Bonami, G. Cornuéjols, and F. Margot, *On the relative strength of split, triangle and quadrilateral cuts*, Mathematical Programming Ser. A, 126 (2009), pp. 281–314, https://doi.org/10.1007/s10107-009-0281-x.

[8] A. Basu, P. Bonami, G. Cornuéjols, and F. Margot, *Experiments with two-row cuts from degenerate tableaux*, INFORMS Journal on Computing, 23 (2011), pp. 578–590.

[9] A. Basu, M. Campêlo, M. Conforti, G. Cornuéjols, and G. Zambelli, *Unique lifting of integer variables in minimal inequalities*, Mathematical Programming, 141 (2013), pp. 561–576, https://doi.org/10.1007/s10107-012-0560-9, http://dx.doi.org/10.1007/s10107-012-0560-9.

[10] A. Basu, M. Conforti, and M. Di Summa, *A geometric approach to cut-generating functions*, Mathematical Programming, 151 (2015), pp. 153–189.

[11] A. Basu, G. Cornuéjols, and M. Köppe, *Unique minimal liftings for simplicial polytopes*, Mathematics of Operations Research, 37 (2012), pp. 346–355.

[12] A. Basu, G. Cornuéjols, and F. Margot, *Intersection cuts with infinite split rank*, Mathematics of Operations Research, 37 (2012), pp. 21–40.

[13] A. Basu, G. Cornuéjols, and M. Molinaro, *A probabilistic analysis of the strength of the split and triangle closures.*, in IPCO, Springer, 2011, pp. 27–38.

[14] A. Basu, S. Dey, and J. Paat, *How to choose what you lift.* manuscript, 2015.

[15] A. Basu, R. Hildebrand, and M. Köppe, *Light on the infinite group relaxation I: Foundations and taxonomy*, 4OR, 14 (2016), pp. 1–40.

[16] A. Basu, R. Hildebrand, and M. Köppe, *Light on the infinite group relaxation II: Sufficient conditions for extremality, sequences, and algorithms*, 4OR, 14 (2016), pp. 1–25.

[17] A. Basu and J. Paat, *Operations that preserve the covering property of the lifting region*, SIAM Journal on Optimization, 25 (2015), pp. 2313–2333.

[18] M. Campêlo and G. Cornuéjols, *Stable sets, corner polyhedra and the chvátal closure*, Operations Research Letters, 37 (2009), pp. 375–378.

[19] M. Conforti, G. Cornuéjols, A. Daniilidis, C. Lemaréchal, and J. Malick, *Cut-generating functions and s-free sets*, Mathematics of Operations Research, 40 (2014), pp. 276–391.

[20] M. Conforti, G. Cornuéjols, and G. Zambelli, *Corner polyhedra and intersection cuts*, Surveys in Operations Research and Management Science, 16 (2011), pp. 105–120.

[21] M. Conforti, G. Cornuéjols, and G. Zambelli, *A geometric perspective on lifting*, Oper. Res., 59 (2011), pp. 569–577, https://doi.org/10.1287/opre.1110.0916, http://dx.doi.org/10.1287/opre.1110.0916.

[22] M. Conforti, G. Cornuéjols, and G. Zambelli, *Integer programming*, vol. 271, Springer, 2014.

[23] G. Cornuéjols, C. Michini, and G. Nannicini, *How tight is the corner relaxation? insights gained from the stable set problem*, Discrete Optimization, 9 (2012), pp. 109–121.

[24] A. Del Pia, *On the rank of disjunctive cuts*, Mathematics of Operations Research, 37 (2012), pp. 372–378.

[25] A. Del Pia, C. Wagner, and R. Weismantel, *A probabilistic comparison of the strength of split, triangle, and quadrilateral cuts*, Operations Research Letters, 39 (2011), pp. 234–240.

[26] A. Del Pia and R. Weismantel, *Relaxations of mixed integer sets from lattice-free polyhedra*, 4OR, 10 (2012), pp. 221–244, https://doi.org/10.1007/s10288-012-0198-8, http://dx.doi.org/10.1007/s10288-012-0198-8.

[27] S. S. Dey, A. Lodi, A. Tramontani, and L. A. Wolsey, *Experiments with two row tableau cuts*, in Integer Programming and Combinatorial Optimization, Springer, 2010, pp. 424–437.

[28] S. S. Dey and L. A. Wolsey, *Composite lifting of group inequalities and an application to two-row mixing inequalities*, Discrete Optim., 7 (2010), pp. 256–268, https://doi.org/10.1016/j.disopt.2010.06.001, http://dx.doi.org/10.1016/j.disopt.2010.06.001.

[29] S. S. Dey and L. A. Wolsey, *Two row mixed-integer cuts via lifting*, Math. Program., 124 (2010), pp. 143–174, https://doi.org/10.1007/s10107-010-0362-x, http://dx.doi.org/10.1007/s10107-010-0362-x.

[30] F. Eisenbrand, *Integer programming and algorithmic geometry of numbers*, in 50 Years of Integer Programming 1958–2008, M. Jünger, T. Liebling, D. Naddef, W. Pulleyblank, G. Reinelt, G. Rinaldi, and L. Wolsey, eds., Springer-Verlag, 2010.

[31] D. G. Espinoza, *Computing with multi-row gomory cuts*, Operations Research Letters, 38 (2010), pp. 115–120.

[32] R. Fukasawa, L. Poirrier, and A. S. Xavier, *Intersection cuts for single row corner relaxations.* http://www.optimization-online.org/DB_FILE/2016/05/5443.pdf, 2016.

[33] R. Fukasawa, L. Poirrier, and A. S. Xavier, *The (not so) trivial lifting in two dimensions.* http://www.math.uwaterloo.ca/ axavier, 2016.

[34] R. E. Gomory, *Some polyhedra related to combinatorial problems*, Linear Algebra and its Applications, 2(4) (1969), pp. 451–558.

[35] R. E. Gomory and E. L. Johnson, *Some continuous functions related to corner polyhedra, I*, Mathematical Programming, 3 (1972), pp. 23–85, https://doi.org/10.1007/BF01585008,

http://dx.doi.org/10.1007/BF01585008.
[36] R. E. Gomory and E. L. Johnson, *Some continuous functions related to corner polyhedra, II*, Mathematical Programming, 3 (1972), pp. 359–389, https://doi.org/10.1007/BF01585008, http://dx.doi.org/10.1007/BF01585008.
[37] Q. He, S. Ahmed, and G. L. Nemhauser, *A probabilistic comparison of split and type 1 triangle cuts for two-row mixed-integer programs*, SIAM Journal on Optimization, 21 (2011), pp. 617–632.
[38] E. L. Johnson, *On the group problem for mixed integer programming*, Mathematical Programming Study, 2 (1974), pp. 137–179.
[39] Q. Louveaux and L. Poirrier, *An algorithm for the separation of two-row cuts*, Mathematical Programming, 143 (2014), pp. 111–146, https://doi.org/10.1007/s10107-012-0597-9.
[40] Q. Louveaux, L. Poirrier, and D. Salvagnin, *The strength of multi-row models*, Mathematical Programming Computation, 7 (2015), pp. 113–148.
[41] L. Lovász, *Geometry of numbers and integer programming*, in Mathematical Programming: State of the Art, M. Iri and K. Tanabe, eds., Mathematical Programming Society, 1989, pp. 177–201.
[42] L. Poirrier, *Multi-row approaches to cutting plane generation*, Ph.D. thesis, University of Liege, Belgium, (2012).
[43] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, New Jersey, 1970.