When Lift-and-Project Cuts are Different

Egon Balas[1] and Thiago Serra[2]

[1]Carnegie Mellon University
[2]Bucknell University

Abstract

In this paper, we present a method to determine if a lift-and-project cut for a mixed-integer linear program is irregular, in which case the cut is not equivalent to any intersection cut from the bases of the linear relaxation. This is an important question due to the intense research activity for the past decade on cuts from multiple rows of simplex tableau as well as on lift-and-project cuts from non-split disjunctions. While it is known since Balas and Perregaard (2003) that lift-and-project cuts from split disjunctions are always equivalent to intersection cuts and consequently to such multi-row cuts, Balas and Kis (2016) have recently shown that there is a necessary and sufficient condition in the case of arbitrary disjunctions: a lift-and-project cut is regular if, and only if, it corresponds to a regular basic solution of the Cut Generating Linear Program (CGLP). This paper has four contributions. First, we state a result that simplifies the verification of regularity for basic CGLP solutions from Balas and Kis (2016). Second, we provide a mixed-integer formulation that checks whether there is a regular CGLP solution for a given cut that is regular in a broader sense, which also encompasses irregular cuts that are implied by the regular cut closure. Third, we describe a numerical procedure based on such formulation that identifies irregular lift-and-project cuts. Finally, we use this method to evaluate how often lift-and-project cuts from simple t-branch split disjunctions are irregular, and thus not equivalent to multi-row cuts, on 74 instances of the MIPLIB benchmarks.

1 Introduction

Many techniques to generate cutting planes for a Mixed-Integer Linear Program (MILP) are equivalent to one another under certain conditions. Since some are more general and usually more expensive computationally, it is important to determine if and when they generate cuts that others cannot. In this paper, we introduce a technique to verify if a lift-and-project cut (Balas et al., 1993, 1996) from an arbitrary disjunction does not correspond to a standard intersection cut (Balas, 1971) and analyze computational results on several instances for insights. This study is particularly relevant due to the recent research activity

around intersection cuts from multiple rows of the simplex tableau since their introduction by Andersen et al. (2007), which in the case of q rows are equivalent to disjunctive cuts from a 2^q-term disjunction (Balas and Qualizza, 2013). More specifically, it helps us find out about the converse: how often lift-and-project cuts from multi-term disjunctions cannot be directly obtained as intersection cuts from solutions of a simplex tableau, and therefore are different. We test when these different cuts – which are denoted as irregular – are found in practice, what may be inducing their occurrence, and how strong they are.

We can obtain a lift-and-project cut by solving a Cut Generating Linear Program (CGLP), which defines valid inequalities for an MILP relaxation consisting of a disjunctive program (Balas, 1979, 1998). Such disjunctive programs are often unions of disjoint polyhedra that exclude the region around a particular solution $x = \bar{x}$ of the Linear Program (LP) relaxation. The most common disjunctive program consist of intersecting the LP relaxation with a *split disjunction* of the form $\{x : \pi x \leq \pi_0\} \vee \{x : \pi x \geq \pi_0 + 1\}$ when $\pi_0 < \pi \bar{x} < \pi_0 + 1$ and no MILP solution is removed by the disjunction. For that case, Balas and Perregaard (2003) have shown that there is a correspondence between lift-and-project cuts from basic CGLP solutions and intersection cuts from basic solutions of the LP relaxation, feasible or not. That entails a more efficient procedure to implicitly solve CGLPs from split disjunctions by pivoting among LP bases, which has been implemented in a number of solvers including CglLandP (Balas and Bonami, 2009) in COIN-OR[1].

The equivalence identified by Balas and Perregaard (2003) associates lift-and-project cuts from a *simple* split disjunction of the form $\{x : x_k \leq 0\} \vee \{x : x_k \geq 1\}$ with Gomory Mixed-Integer cuts (Gomory, 1960) from the row of some simplex tableau, feasible or not, defining the value of x_k in terms of nonbasic continuous variables. Similarly, strengthening those lift-and-project cuts by modularizing the coefficients associated with integer nonbasic variables (Balas and Jeroslow, 1980; Balas et al., 1993) makes them equivalent to GMI cuts from the corresponding row of some simplex tableau that defines x_k in terms of nonbasic variables, some or all of which may be integer-constrained.

More recently, Balas and Kis (2016) have shown that the correspondence between lift-and-project cuts and intersection cuts may not necessarily hold for lift-and-project cuts from arbitrary disjunctions. More specifically, they have proven that it holds if, and only if, there is a basic CGLP solution associated with the cut that maps to an LP basis that corresponds to a standard intersection cut. These basic CGLP solutions mapping to LP bases are called *regular*. A lift-and-project cut is then called *regular* if there exists a corresponding regular basic CGLP solution and *irregular* otherwise.

The elegance and convenience of generating intersection cuts from the simplex tableau has motivated a recent stream of theoretical work on generating cuts from two rows of the simplex tableau (Andersen et al., 2007; Cornuéjols and Margot, 2008) and subsequently more rows (Borozan and Cornuéjols, 2009; Basu et al., 2010), on how these cuts can be strengthened when nonbasic variables are integer (Dey and Wolsey, 2010; Conforti et al.,

---

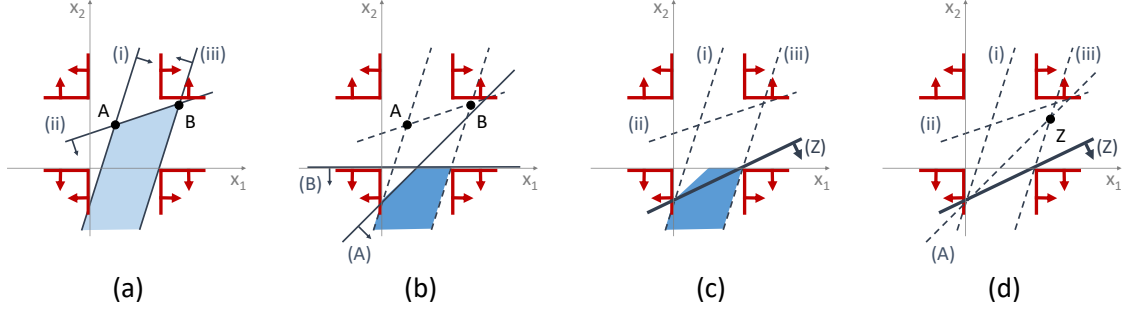[1] Available at http://www.coin-or.org

2

Figure 1: (a) LP relaxation and a 2-branch split disjunction; (b) Non-dominated intersection cuts; (c) Irregular cut dominating all intersection cuts; and (d) Additional inequality making previous cut regular.

2011a; Basu et al., 2013; Fukasawa et al., 2016), and several other variants. The reader is referred to Conforti et al. (2011b) and Basu et al. (2015) for a broader review of this line of work, which has been accompanied by extensive computational experimentation (Espinoza, 2008; Basu et al., 2011; Dey et al., 2014; Louveaux et al., 2015).

However, there are other ways in which one can exploit that more than one integer variable is fractional in $\bar{x}$. For example, we can generate lift-and-project cuts using disjunctions with more than two terms, and those may yield irregular cuts instead. A natural generalization of the commonly used split disjunction is defined by Li and Richard (2008) as the *t-branch split disjunction*

$$\bigvee_{S \subseteq \{1,2,\ldots,t\}} \left\{ x : \pi^i x \leq \pi_0^i, \text{ if } i \in S; \ \pi^i x \geq \pi_0^i + 1, \text{ if } i \notin S \right\},$$

of which the cross disjunction (Dash et al., 2012) corresponds to the special case where $t = 2$. Dash et al. (2014) observed that the resulting cuts close a substantially larger gap in comparison to split cuts. Furthermore, Andersen et al. (2005) as well as Kis (2014) have shown examples of disjunctive cuts that do not correspond to intersection cuts.

Figure 1 illustrates another such case. In Figure 1a, we have an LP relaxation defined by inequalities (i) $6x_1 - 2x_2 \geq 1$, (ii) $2x_1 - 6x_2 \geq -3$, and (iii) $-3x_1 + x_2 \geq -3$ in lighter blue and a 2-branch split disjunction $\{x : x_1 \geq 1, x_2 \geq 1\} \vee \{x : x_1 \leq 0, x_2 \geq 1\} \vee \{x : x_1 \leq 0, x_2 \leq 0\} \vee \{x : x_1 \geq 1, x_2 \leq 0\}$ in red. The LP has basic feasible solutions at points $A = (\frac{3}{8}, \frac{5}{8})$ and $B = (\frac{21}{16}, \frac{15}{16})$. In Figure 1b, we have the non-dominated intersection cuts (A) $2x_1 - 2x_2 \geq 1$ and (B) $x_2 \leq 0$ defining the regular cut closure in the darker blue area. These cuts are respectively associated with the LP bases in points A and B. In Figure 1c, we show another valid inequality (Z) $x_1 - 2x_2 \geq 1$ cuts through the regular cut closure and in fact dominates the intersection cuts from the LP bases within the LP relaxation. While each intersection cut can be obtained with the two inequalities that are satisfied at

3

equality of the corresponding LP basis, the three inequalities are necessary in the case of (Z): we need both inequalities (ii) and (iii) to certify that (Z) does not separate any point at the intersection of the LP relaxation with the first term of the disjunction, and similarly we need inequality (i) in the case of the the third term of the disjunciton. In Figure 1d, we show that (Z) would be an intersection cut from an LP basis at $Z = (\frac{5}{4}, \frac{3}{4})$ if the LP relaxation also includes the valid inequality (A). Hence, we cannot infer that a lift-and-project cut is irregular by only inspecting the CGLP solution from which we obtained the cut, since it could also be generated in other ways.

In this work, we focus on the equivalence with respect to lift-and-project cuts without strengthening, and we present the following contributions. First, we state a result that simplifies the verification of regularity for basic CGLP solutions from Balas and Kis (2016) and show how it extends to CGLP solutions that are not basic in Section 3. Second, we introduce and prove the validity of an MILP that checks whether there is a regular CGLP solution, basic or not, for a given cut in Section 4. Third, we describe a numerical procedure based on such MILP that verifies if a lift-and-project cut is irregular in a stricter sense in Section 5. Finally, we present computational results from 74 benchmark instances in Section 6, and we use these results to analyze what factors may lead to a higher incidence of strictly irregular cuts and how these cuts compare with the other cuts in Section 7.

2   Preliminaries

Let us consider a mixed $0-1$ linear program with rational coefficients

$$\min \Big\{ cx \ : \ Ax \geq b, x \geq 0, \ x_j \in \{0,1\}, \ j = 1, \dots, p \Big\}, \tag{$\mathcal{P}$}$$

where A is an $m \times n$ matrix. Let $\tilde{A}x \geq \tilde{b}$ denote the constraint set of the LP relaxation $P := \{x : Ax \geq b, x \geq 0, x_j \leq 1, j = 1, \dots, p\}$ and let $P_I := \{x : x \in P, x_j \in \{0,1\}, j = 1, \dots, p\}$ denote the feasible set of $(\mathcal{P})$. Hence, $\tilde{A}$ is a $q \times n$ matrix, where $q = m + n + p$. Let $Q := \{1, \dots, q\}$.

Furthermore, let $\bar{x}$ be a basic optimal solution of the LP relaxation of $(\mathcal{P})$, i.e., $\bar{x} = \arg\min\{cx : \tilde{A}x \geq \tilde{b}\}$. If $\bar{x}$ is not feasible for $(\mathcal{P})$, we can define a disjunction $\vee_{t \in T} D^t x \geq d_0^t$ that contains the feasible set of $(\mathcal{P})$ and not $\bar{x}$. For example, if there is a nonempty set $K \subseteq \{1, \dots, p\}$ for which $0 < \bar{x}_k < 1$ for every $k \in K$, then we can define a disjunction of the form

$$\bigvee_{K' \subseteq K} \left(\begin{array}{ll} x_k \geq 1, & k \in K' \\ x_k \leq 0, & k \in K \setminus K' \end{array} \right), \tag{1}$$

which we denote as a *simple t-branch split disjunction*[2], where $t = |K|$. More generally, we can generate lift-and-project cuts by intersecting sets such as (1) with the linear relaxation

---

[2]Unless noted otherwise, we will use t to index terms of a disjunctive program in general form instead of parameterizing a t-branch split disjunction. We will use a set K to denote a particular simple t-branch split disjunction.

of ($\mathcal{P}$). More generally, we have a disjunctive program of the form

$$\bigvee_{t \in T} \begin{pmatrix} \tilde{A}x \geq \tilde{b} \\ D^t x \geq d_0^t \end{pmatrix}, \tag{2}$$

where, without loss of generality, D^t is a $r \times n$ matrix. The classic formulation for the Cut Generating Linear Program (CGLP) used to find a lift-and-project cut among valid inequalities for (2) separating $\bar{x}$, which excludes some dominated inequalities, is as follows:

$$\min \ \alpha \bar{x} \quad - \beta \tag{3}$$

$$\alpha \qquad\qquad -u^t \tilde{A} \qquad -v^t D^t = 0, \quad t \in T \tag{4}$$

$$\beta \qquad -u^t \tilde{b} \qquad -v^t d_0^T = 0, \quad t \in T \tag{5}$$

$$\sum_{t \in T} \sum_{i=1}^{q} u_i^t + \sum_{t \in T} \sum_{i=1}^{r} v_i^t = 1 \tag{6}$$

$$u^t, \qquad\qquad v^t \geq 0, \quad t \in T \tag{7}$$

$$\alpha \in \mathbb{R}^n, \quad \beta \in \mathbb{R} \tag{8}$$

In the formulation above, the data consist of $\tilde{A}$, $\tilde{b}$, D^t and d_0^t for $t \in T$. The variables are α, β, u^t and v^t for $t \in T$. If $\alpha \bar{x} - \beta < 0$, then $\alpha x \geq \beta$ is a valid inequality separating $\bar{x}$.

For every choice of multipliers $\{v^t\}_{t \in T}$ such that $v^t \geq 0$ and $v^t \neq 0$, where v^t is a vector in which each element corresponds to an inequality of $D^t x \geq d_0^t$, we can define a polyhedron $S\left(\{v^t\}_{t \in T}\right) := \{x : (v^t D^t)x \leq v^t d_0^t\}$. If $(v^t D^t)\bar{x} < v^t d_0^t$ for each $t \in T$, then such polyhedron contains $\bar{x}$ as an interior point but no point in P_I, and thus it is possible to use polyhedron $S\left(\{v^t\}_{t \in T}\right)$ to derive an intersection cut separating $\bar{x}$ from P_I as discussed next.

If we assume, without loss of generality, that the upper bounds on the binary variables are contained in $Ax \geq b$, then we extend x with surplus variables of the form $x_{n+i} = \sum_{j=1}^{n} a_{ij} x_j - b_i$ for $i = 1, \ldots, m$ and define the following LP cone $C(J)$ from each basic solution $x(J)$ of P:

$$x_i = \bar{a}_{i0} - \sum_{j \in J} \bar{a}_{ij} x_j, \qquad i \in I \tag{9}$$

$$x_i \geq 0, \qquad i \in J \tag{10}$$

where I is the index set of basic variables and J of the nonbasic variables. Cone $C(J)$ has an extreme ray $r^j(J)$ corresponding to each nonbasic variable $x_j, j \in J$, with $r_j^j(J) = 1$, $r_i^j(J) = -\bar{a}_{ij}$ for $i \in I$, and $r_i^j(J) = 0$ for $i \in J \setminus \{j\}$. If there is a convex set S containing $x(J)$ but no feasible integer point in its interior, which we denote as P_I-free, we can define the intersection cut (Balas, 1971)

$$\sum_{j \in J} \frac{1}{\lambda_j^*} x_j \geq 1 \tag{11}$$

5

separating $x(J)$ from P_I, where λ_j^* defines the point at which each ray $x(J) + \lambda_j r_j(J)$ intersects the boundary of S. If some ray never intersects the boundary, then the corresponding coefficient of the intersection cut is zero instead of λ_j^* inversed.

If a lift-and-project cut $\alpha x \geq \beta$ separating $\bar{x}$ is obtained from a basic CGLP solution defined on a split disjunction $\{x : \pi x \leq \pi_0\} \vee \{x : \pi x \geq \pi_0 + 1\}$, then the rows of the LP relaxation associated with the basic multipliers in the CGLP solution define the LP basis from which $\alpha x \geq \beta$ can be obtained as an intersection cut using the P_I-free set $\{x : \pi_0 \leq \pi x \leq \pi_0 + 1\}$. In this particular case, Balas and Perregaard (2003) have shown that there are exactly n basic multipliers in u associated with distinct rows of the LP relaxation.

For arbitrary disjunctions, the lift-and-project cuts that are equivalent to intersection cuts are always associated with at least one basic CGLP solution that generalizes the structure found in the case of split disjunctions. The following results, which are adapted from Balas and Kis (2016), present the necessary and sufficient conditions for such equivalence:

Theorem 1 (Balas and Kis (2016), Th. 10)**.** *Let* $(\bar{\alpha}, \bar{\beta}, \{\bar{u}^t, \bar{v}^t\}_{t \in T})$ *be a basic feasible solution to* (4)–(8) *such that* $\sum_{i=1}^{r} \bar{v}_i^t > 0$ *for all* $t \in T$. *If there exists a nonsingular* $n \times n$ *submatrix* $\tilde{A}_J$ *of* $\tilde{A}$ *such that* $\bar{u}_j^t = 0$ *for all* $j \notin J$ *and* $t \in T$, *then the lift-and-project cut* $\bar{\alpha} x \geq \bar{\beta}$ *is equivalent to the intersection cut* $\pi x_J \geq 1$ *from the set*

$$S(\bar{v}) := \{x \in \mathbb{R}^n : (\bar{v}^t D^t)x \leq \bar{v}^t d_0^t, t \in T\}$$

and the LP simplex tableau with nonbasic set J.

Theorem 2 (Balas and Kis (2016), Th. 12)**.** *Let* $\bar{w} = (\bar{\alpha}, \bar{\beta}, \{\bar{u}^t, \bar{v}^t\}_{t \in T})$ *be a basic feasible solution to* (4)–(8) *such that* $\sum_{i=1}^{r} \bar{v}_i^t > 0, t \in T$.

If $\bar{w}$ *does not satisfy the (sufficient) condition of Theorem 1, and there is no basic feasible solution* $\tilde{w}$ *to* (4)–(8) *with* $(\tilde{\alpha}, \tilde{\beta}) = \mu(\bar{\alpha}, \bar{\beta})$ *for some* $\mu > 0$ *that satisfies the condition of Theorem 1, then there exists no intersection cut from any member of the family of polyhedra*

$$S(v) := \{x \in \mathbb{R}^n : (v^t D^t)x \leq v^t d_0^t, t \in T\}$$

where $v \geq 0, v \neq 0$, *equivalent to* $\bar{\alpha}\bar{x} \geq \bar{\beta}$.

In other words, a cut $\bar{\alpha} x \geq \bar{\beta}$ from a regular basic solution $(\bar{\alpha}, \bar{\beta}, \{\bar{u}^t, \bar{v}^t\}_{t \in T})$ is equivalent to the intersection cut from the LP cone associated with the cobasis indexed by J and the P_I-free convex set defined by $\{x : (\bar{v}^t D^t)x \leq \bar{v}^t d_0^t, t \in T\}$. More specifically, a positive CGLP multiplier $\bar{u}_i^t$ for some row i of $Ax \geq b$ for any $t \in T$ maps the corresponding surplus variable s_i as nonbasic in the LP, a positive multiplier $\bar{u}_{m+j}^t$ for the bound $x_j \geq 0$ maps x_j as nonbasic at the lower bound in the LP, and a positive CGLP $\bar{u}_{m+p+j}^t$ for the bound $x_j \leq 1$ maps x_j as nonbasic at the upper bound in the LP. The inequality $\bar{\alpha} x \geq \bar{\beta}$ only cuts

off part of P if $\sum_{i=1}^{r} \bar{v}_i^t > 0$ for each $t \in T$, since otherwise $\bar{\alpha}x \geq \bar{\beta}$ is implied by $\tilde{A}x \geq \tilde{b}$, as shown in Lemma 1 of Balas and Perregaard (2003). Furthermore, the sufficient condition for the regularity of $\bar{w}$ is also necessary for $\bar{\alpha}x \geq \bar{\beta}$, i.e., if the condition is not met by any other CGLP basis, then the cut is irregular.

The following definition for what constitutes a regular basic CGLP solution and regular lift-and-project cut is also adapted from Balas and Kis (2016):

Definition 1 (Regularity of bases and cuts (Balas and Kis, 2016))**.** *A feasible basis for the CGLP system* (4)–(8) *and the associated solution will be called* regular *if the basis satisfies the condition of Theorem 1, and* irregular *otherwise. A cut defined by an irregular solution w is* irregular*, unless there exists a regular solution w' with the same (α, β)-component – upon scaling by a positive multiplier – as that of w, in which case the cut is* regular*.*

3   Regularity of CGLP Solutions

The next Theorem gives a simple criterion for deciding whether a basic CGLP solution $\bar{w}$ is regular or not.

Theorem 3. *For a basic CGLP solution $\bar{w} = (\bar{\alpha}, \bar{\beta}, \{\bar{u}^t, \bar{v}^t\}_{t \in T})$, let $\tilde{A}_N$ be the $|N| \times n$ submatrix of $\tilde{A}$ whose rows are indexed by $N(\bar{u}) := \{j \in Q : \bar{u}_j^t > 0 \text{ for some } t \in T\}$. Then $\bar{w}$ is a regular solution if, and only if, $\tilde{A}_N$ is of full row rank.*

Proof. Proof. *Sufficiency.* Assume $\operatorname{rank}(\tilde{A}_N) = |N|$. Then $|N| \leq n$. We show that in this situation $\bar{w}$ is regular.

Case 1: $|N| = n$. Then $\tilde{A}_N$ is an $n \times n$ nonsigular submatrix of $\tilde{A}$ such that $u_j^t = 0$ for all $j \notin N$ and all $t \in T$, i.e., $\bar{w}$ satisfies the condition of Theorem 1.

Case 2: $|N| < n$. Then $\tilde{A}$ has $n - |N|$ rows $\tilde{A}_j$ with $u_j^t = 0$ which can be added to $\tilde{A}_N$ in order to form an $n \times n$ nonsingular matrix $\tilde{A}_{N'}$ since $\tilde{A}$ contains I_n. Substituting $\tilde{A}_{N'}$ for $\tilde{A}_N$ then reduces this case to Case 1.

Necessity. Assume $\operatorname{rank}(\tilde{A}_N) < |N|$. We show that in this case $\bar{w}$ is irregular. In particular, any $n \times n$ nonsingular submatrix $\tilde{A}_J$ of $\tilde{A}$ has among its rows at most $\operatorname{rank}(\tilde{A}_N)$ rows of $\tilde{A}_N$, thus leaving $|N| - \operatorname{rank}(\tilde{A}_N)$ rows j such that $\bar{u}_j^t > 0$ for some $t \in T$ outside of $\tilde{A}_J$. Therefore no such $\tilde{A}_J$ meets the condition of Theorem 1, hence $\bar{w}$ is irregular. $\square$

In Figure 1c, note that the irregular cut (Z) can only be obtained by using the parallel inequalities (i) and (iii), for which reason the submatrix is not of full row rank.

Corollary 4. *If a CGLP solution $\bar{w} = (\bar{\alpha}, \bar{\beta}, \{\bar{u}^t, \bar{v}^t\}_{t \in T})$ is not basic but satisfies the conditions of Theorem 3 for regularity, then $\bar{\alpha}x \geq \bar{\beta}$ is valid for the closure of regular cuts.*

Proof. Proof. If $\bar{w}$ is not basic, then we can describe it as a proper convex combination of a set of basic CGLP solutions. Let these solutions be indexed by a given set $\mathcal{B}$, so that $(\bar{\alpha}, \bar{\beta}, \{\bar{u}^t, \bar{v}^t\}_{t \in T}) = \sum_{b \in \mathcal{B}} \lambda_b \left(\tilde{\alpha}^b, \tilde{\beta}^b, \left\{ \tilde{u}^{t^b}, \bar{v}^{t^b} \right\}_{t \in T} \right)$, $\lambda > 0$, and $\sum_{b \in \mathcal{B}} \lambda_b = 1$.

Note that $\tilde{u}_j^{t^b} > 0$ implies that $\bar{u}_j^t > 0$, and thus $N(\tilde{u}^b) \subseteq N(\bar{u})$ for all $b \in \mathcal{B}$. Since the submatrix of $\tilde{A}$ on the rows of $N(\bar{u})$ is of full row rank, then there exists a set N such that $|N| = n$ and $N(\bar{u}) \subseteq N$ for which $\text{rank}(\tilde{A}_N) = n$, and thus $N(\tilde{u}^b) \subseteq N$ for all $b \in \mathcal{B}$. That implies that all inequalities of the form $\tilde{\alpha}^b x \geq \tilde{\beta}^b$ for each $b \in \mathcal{B}$ define intersection cuts from a same basis, and thus each of those is regular. $\qquad\square$

Motivated by Corollary 4, we consider for the rest of the paper a broader definition of regularity for CGLP solutions, which conversely implies a stricter definition for cut irregularity that disregards certain cuts that are implied by the set of regular cuts:

Definition 2 (Extended Regularity). *A feasible solution for the CGLP system* (4)–(8) *will be called* extended regular *if it satisfies all conditions of Theorem 3 other than being basic. A lift-and-project cut $\alpha x \geq \beta$ is* strictly irregular *if there is no extended regular solution with the same (α, β)-component upon scaling by a positive multiplier.*

Thus deciding whether a basic CGLP solution $\bar{w}$ is regular or not is straightforward. Furthermore, any regular CGLP solution suffices to prove that a given cut is not strictly irregular. Next we examine how to use that to find strictly irregular cuts.

4   Regularity of Cuts from CGLP Solutions

Given an irregular CGLP solution $\bar{w} = (\bar{\alpha}, \bar{\beta}, \{\bar{u}^t, \bar{v}^t\}_{t \in T})$, we define a mixed-integer program based on the CGLP to establish whether there is an extended regular CGLP solution $\tilde{w} = (\tilde{\alpha}, \tilde{\beta}, \{\tilde{u}^t, \tilde{v}^t\}_{t \in T})$ such that $(\tilde{\alpha}, \tilde{\beta}) = \theta(\bar{\alpha}, \bar{\beta})$ for some $\theta > 0$. In comparison to the CGLP formulation, we add a variable $\theta \in [0, 1]$ and restrict the value of (α, β) to $\theta(\bar{\alpha}, \bar{\beta})$. Furthermore, we remove the normalization constraint (6) and introduce a binary upper bounding variable δ_j for each u_j in order to model the set N of indices $j \in Q$ such that $u_j^t > 0$ for some $t \in T$. We embed Theorem 3 by restricting the size of such set to at most n and requiring that the submatrix $\tilde{A}_N$ to be of full row rank. We denote the resulting formulation as the Regular Cut Verifier MILP (RCV-MILP):

$$\max \quad \theta \tag{12}$$

$$\theta\bar{\alpha} \qquad -u^t\tilde{A} - v^t D^t = 0, \qquad\qquad t \in T \tag{13}$$

$$\theta\bar{\beta} \qquad -u^t\tilde{b} - v^t d_0^t = 0, \qquad\qquad t \in T \tag{14}$$

$$\delta_j \qquad -u_j^t \qquad \geq 0, \qquad j \in Q, \ t \in T \tag{15}$$

$$\sum_{j \in Q} \delta_j \qquad\qquad \leq n \tag{16}$$

$$\sum_{j \in N} \delta_j \qquad\qquad \leq \operatorname{rank}(\tilde{A}_N), \qquad N \subseteq Q \tag{17}$$

$$u^t, v^t \qquad \geq 0, \qquad\qquad t \in T \tag{18}$$

$$\delta_j \qquad \in \{0, 1\}, \qquad j \in Q \tag{19}$$

$$\theta \qquad \in [0, 1] \tag{20}$$

The data consist of $\bar{\alpha}$, $\bar{\beta}$, $\tilde{A}$, the rank of all submatrices of $\tilde{A}$, $\tilde{b}$, D^t and d_0^t for $t \in T$. The variables are θ, δ, u^t and v^t for $t \in T$. If $\theta > 0$, then $\bar{\alpha}x \geq \bar{\beta}$ is a regular cut.

Some comments regarding RCV-MILP are in order. First, constraints (15) and (19) define an implicit normalization constraint $\|u\|_\infty \leq 1$, which may prevent us from finding the cut in the same scale. Hence, variable θ is necessary even though normalization (6) is removed in comparison to the CGLP. Since it is not immediate that the implicit normalization guarantees that RCV-MILP is always bounded, we set an upper limit of 1 to θ. Second, the equivalence between a RCV-MILP solution $(\check{\theta}, \check{\delta}, \{\check{u}^t, \check{v}^t\}_{t \in T})$ and a regular CGLP solution denoting a cut equivalent to $\bar{\alpha}x \geq \bar{\beta}$ is not direct. Instead of simply stating a corresponding CGLP solution $(\check{\theta}\bar{\alpha}, \check{\theta}\bar{\beta}, \{\check{u}^t, \check{v}^t\}_{t \in T})$, we may need to first scale the RCV-MILP solution if $\sum_{t \in T} \sum_{i=1}^r \check{u}_i^t + \sum_{t \in T} \sum_{i=1}^r \check{v}_i^t \neq 1$ in order to satisfy normalization (6). Such scaling is done a number of times in the next proof. Last, for each subset $N \subseteq Q$ we assume that the rank of $\tilde{A}_N$ is given as an input. However, the number of subsets of Q, and consequently the number of rows due to constraint (17), can be very large. In Section 5, we address that by iteratively adding to the formulation only the relevant subsets of Q and computing the rank of the corresponding submatrices of $\tilde{A}$.

The next result proves the validity of RCV-MILP. In what follows, we keep denoting the set of rows with positive multipliers as $N(\bar{u})$. Furthermore, let $\delta(\bar{u})$ be a vector in which $\delta_i = 1$ if $\bar{u}_i^t > 0$ for some $t \in T$ and $\delta_i = 0$ otherwise.

Theorem 5. *Let* $\bar{w} = (\bar{\alpha}, \bar{\beta}, \{\bar{u}^t \bar{v}^t\}_{t \in T})$ *be a basic optimal solution of CGLP and let* $(\check{\theta}, \check{\delta}, \{\check{u}^t, \check{v}^t\}_{t \in T})$ *be an optimal solution of RCV-MILP for cut* $\bar{\alpha}x \geq \bar{\beta}$. *Then the cut is not strictly irregular if, and only if, $\check{\theta} > 0$.*

Proof. Proof. If $\bar{w}$ is a regular CGLP solution, then $(1, \delta(\bar{u}), \{\bar{u}^t, \bar{v}^t\}_{t \in T})$ is an optimal solution of RCV-MILP. First, $\theta = 1$ implies that constraints (4) and (5) are equivalent to

9

(13) and (14), whereas constraints (7) and (18) are the same. Second, normalization (6) implies that $\bar{u} \leq 1$ and thus $\delta(\bar{u})$ satisfies constraints (15) and (19). Last, constraints (16) and (17) are satisfied since $\bar{w}$ is regular.

For the rest of the proof, we assume that the CGLP solution $\bar{w}$ is irregular.

Suppose that the cut is not strictly irregular, and thus there is an extended regular CGLP solution $\tilde{w} = (\tilde{\alpha}, \tilde{\beta}, \{\tilde{u}^t, \tilde{v}^t\}_{t \in T})$ for which $(\bar{\alpha}, \bar{\beta}) = \lambda(\tilde{\alpha}, \tilde{\beta})$ for some $\lambda > 0$. If $\lambda \leq 1$, then $(\lambda, \delta(\tilde{u}), \{\tilde{u}^t, \tilde{v}^t\}_{t \in T})$ is feasible for RCV-MILP and thus $\check{\theta} \geq \lambda > 0$. If $\lambda > 1$, we can divide the same values by λ to obtain another feasible CGLP solution $\hat{w} = \left(\frac{1}{\lambda}\tilde{\alpha}, \frac{1}{\lambda}\tilde{\beta}, \left\{ \frac{1}{\lambda}\tilde{u}^t, \frac{1}{\lambda}\tilde{v}^t \right\}_{t \in T} \right) = \left(\bar{\alpha}, \bar{\beta}, \left\{ \frac{1}{\lambda}\tilde{u}^t, \frac{1}{\lambda}\tilde{v}^t \right\}_{t \in T} \right)$. The construction of a RCV-MILP solution when $\lambda \leq 1$ applies to $\hat{w}$, thereby implying that $\left(1, \delta(\tilde{u}), \left\{ \frac{1}{\lambda}\tilde{u}^t, \frac{1}{\lambda}\tilde{v}^t \right\}_{t \in T} \right)$ is optimal for RCV-MILP and thus $\check{\theta} = 1$.

Finally, suppose that the cut is strictly irregular, and thus there is no extended regular CGLP solution corresponding to cut $\bar{\alpha}x \geq \bar{\beta}$. Let us suppose, for contradiction, that there is a RCV-MILP solution $(\check{\theta}, \check{\delta}, \{\check{u}^t, \check{v}^t\}_{t \in T})$ with $\check{\theta} > 0$. Let $\sigma := \sum_{t \in T} \sum_{i=1}^{r} \check{u}_i^t + \sum_{t \in T} \sum_{i=1}^{r} \check{v}_i^t$. Since the cut separates $\bar{x}$ and $\check{\theta} > 0$, then $(\bar{\alpha}, \bar{\beta}) \neq 0$ and thus $\sigma > 0$. Hence, if we divide the multipliers $\{\check{u}^t, \check{v}^t\}_{t \in T}$ by σ, then normalization (6) is satisfied and $\left(\frac{\check{\theta}}{\sigma}\bar{\alpha}, \frac{\check{\theta}}{\sigma}\bar{\beta}, \left\{ \frac{1}{\sigma}\check{u}^t, \frac{1}{\sigma}\check{v}^t \right\}_{t \in T} \right)$ is a feasible extended regular CGLP solution: a contradiction. □

5   Numerical Procedure

Next we describe a numerical procedure to identify strictly irregular lift-and-project cuts, which addresses two issues with using RCV-MILP directly.

First, finite numerical precision may lead to rounding errors and cause false negatives, thus overcounting the number of strictly irregular cuts. We address that by adding a relative tolerance parameter ε on the coefficients of the cut. For example, if we want to determine if a cut $2x_1 + 0.3x_2 \geq 10$ is regular for $\varepsilon = 0.0001$, then we look for valid inequalities on each term where $\alpha_1 \in \theta[1.9998, 2.0002]$, $\alpha_2 \in \theta[0.29997, 0.30003]$, and $\beta \in \theta[9.999, 10.001]$. We therefore avoid false negatives (often described as type II errors) at the price of tolerating false positives (often described as type I errors) when testing which cuts are regular. This choice is intentional, since we are mainly interested in knowing which lift-and-project cuts are not regular. Furthermore, if ε is sufficiently small, misclassifications are very unlikely.

Second, the number of subsets of Q can be very large and many of those subsets might be irrelevant. We can address that by defining a set $\mathcal{Q}$ of subsets of Q and then adding elements to this set as needed.

Hence, we define the Iterative RCV-MILP (IRCV-MILP):

$$\max\ \theta$$

$$-\theta\varepsilon \leq \theta\bar{\alpha} - u^t\tilde{A} - v^t D^t \leq \theta\varepsilon, \qquad\qquad t \in T$$

$$-\theta\varepsilon \leq \theta\bar{\beta} - u^t\tilde{b} - v^t d_0^t \leq \theta\varepsilon, \qquad\qquad t \in T$$

$$\delta_j - u_j^t \geq 0, \qquad\qquad j \in Q, t \in T$$

$$\sum_{j\in Q}\delta_j \leq n$$

$$\sum_{j\in N}\delta_j \leq \mathrm{rank}(\tilde{A}_N), \qquad\qquad N \subseteq Q$$

$$u^t, v^t \geq 0, \qquad\qquad t \in T$$

$$\delta_j \in \{0,1\}, \qquad\qquad j \in Q$$

$$\theta \in [0,1]$$

Algorithm 1 uses IRCV-MILP to determine if a cut is regular, subject to false positives only. It finishes in finite time since each repetition of the loop corresponds to adding a different subset N to $\mathcal{Q}$. In the unlikely worst case, Algorithm 1 terminates when all subsets of Q have been added to $\mathcal{Q}$.

Algorithm 1 Checks if there is a regular CGLP solution for a given cut
---
1: **function** IsCutRegular($\bar{\alpha}, \bar{\beta}, \{\bar{u}^t, \bar{v}^t\}_{t\in T},\ \varepsilon$)
2:     $N \leftarrow N(\bar{u})$
3:     **if** $\mathrm{rank}(\tilde{A}_N) = |N|$ **then**
4:         **return** True   ▷ Original CGLP solution is regular
5:     **else**
6:         $\mathcal{Q} \leftarrow \emptyset$
7:         **loop**
8:             Get optimal solution $(\check{\theta}, \check{\delta}, \{\check{u}^t, \check{v}^t\}_{t\in T})$ of IRCV-MILP
9:             $N \leftarrow N(\check{u})$
10:             **if** $\bar{\theta} = 0$ **then**
11:                 **return** False   ▷ There is no regular CGLP solution
12:             **else if** $\mathrm{rank}(\tilde{A}_N) < |N|$ **then**
13:                 $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{N\}$ ▷ Loop has to be repeated
14:             **else**
15:                 **return** True   ▷ Found regular CGLP solution
16:             **end if**
17:         **end loop**
18:     **end if**
19: **end function**

We can reduce the number of loop repetitions by preventing combinations of inequalities corresponding to parallel hyperplanes across different terms of the disjunction. For

example, if rows j_1 and j_2 correspond to $x_i \geq 0$ and $x_i \leq 1$, then we can add the following inequality to IRCV-CGLP:

$$\delta_{j_1} + \delta_{j_2} \leq 1 \tag{21}$$

6   Computational Experiments

We have run experiments to find strictly irregular lift-and-project cuts among the first round of cuts generated for 74 instances from the MIPLIB 2, 3, and 2003 benchmarks (Bixby et al., 1992, 1998; Achterberg et al., 2006). For each of those instances, we found an optimal solution $\bar{x}$ of the LP relaxation and generated a cut using the CGLP from each disjunction of the form (1), i.e. $\bigvee_{K' \subseteq K} \{x_k \geq 1 \text{ if } k \in K'; \ x_k \leq 0 \text{ otherwise}\}$, where $0 < \bar{x}_k < 1$ for each $k \in K$, with $|K| = 2$ for all instances as well as $|K| = 3$ and $|K| = 4$ for smaller ones. Since the verification can be computationally expensive, the experiments are restricted to instances with at most 10,000 nonzeroes and each verification was interrupted if inconclusive after a predefined number of steps or time, which are both detailed later. All code is written in C++ (gcc 4.8.2) and ran in Ubuntu 14.04.1 LTS on a machine with 48 Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz processors and 32 GB of RAM. No more than two copies were run in parallel and each was restricted to 10 GB of RAM. The formulations were solved with CPLEX version 12.6.3 and matrix ranks were computed with Eigen[3]. Finally, we have run Algorithm 1 to determine if the cut is strictly irregular with $\varepsilon = 0.0001$ and constraints of the form (21) for the bounds of each variable x_i, $i = 1, \ldots, p$. For brevity, we denote strictly irregular cuts as irregular cuts and the remaining cuts implied by the closure of regular cuts as regular cuts in the tables and figures.

The most extensive experiments were run on 45 instances, where we counted the number of times that the loop of Algorithm 1 is repeated and set a time limit of 1 hour to interrupt the verification for each cut. We have chosen the subset of instances with at most 2,000 nonzeroes, 150 integer variables, or 50 rows; and we have also included larger instances from families of instances that nearly met those conditions, namely p0291, misc03, misc07, and pp08a. For instances with at most 20 fractional values in $\bar{x}$ for integer variables, we generated and tested cuts with $|K| = 2$ to $|K| = 4$. For instances with at most 30 fractional values, we generated and tested cuts with $|K| = 2$ and $|K| = 3$. For other instances with at most 50 fractional values and pp08a, we only generated and tested cuts with $|K| = 2$. Some instances are discarded or only the results for less disjunctions are reported in cases where too many verifications timed out or the computer program ran out of resources.

We also verified if the cuts that are obtained with basic regular CGLP solutions map to split cuts. For a disjunction defined by K, we check if only one element in v^t is positive for each $t \in T$ and if those positive multipliers correspond to inequalities on a same variable x_k, $k \in K$, in which case the convex set associated with the intersection cut is $\{x : 0 \leq x_k \leq 1\}$.

---

[3]Available at http://eigen.tuxfamily.org

The results for 36 instances having cuts with $|K| = 2$ and $|K| = 3$ are found in Table 1. Additional results for the 22 instances having cuts with $|K| = 4$ are found in Table 2. We also compare the incidence of irregular CGLP solutions and cuts with $|K| = 2$ to $|K| = 4$ for those 22 instances in Table 3. The results for the remaining 9 instances having cuts with $|K| = 2$ are found in Table 4. Since the number of cuts for each instance varies, we summarize a per-instance average of the percentages for each metric, which weighs the results of each instance for equal contribution.

Figure 2 shows the number of cuts identified as regular and irregular by Algorithm 1 according to the order of magnitude of repetitions of the loop upon termination. Figure 3 compares the regularity of cuts from each K of size 3 and 4 with that of cuts from subsets of K of size 2 among the 97% of the cases where no cut for a subset timed out. In other words, for each set $K = \{k_1, \ldots, k_m\}, m \in \{3, 4\}$, that defines a simple m-branch split disjunction from which we obtain a regular or irregular cut, we count how many irregular cuts are obtained from each subset $K' \subset K, |K'| = 2$, which defines a simple 2-branch split disjunction. There are 3 of those subsets of size 2 in K if $|K| = 3$ and 6 if $|K| = 4$. Figure 4 compares the average gap closed and the average Euclidean distance of the cuts generated with respect to the fractional solution $\bar{x}$ on all disjunction sizes where both types of cuts are observed on each of the 45 instances. Figure 5 shows the total gap closed with and without irregular cuts in the cases in which both types of cuts are observed.

Figure 2: Incidence of cuts assessed by Algorithm 1 by the order of repetitions of the loop upon termination.
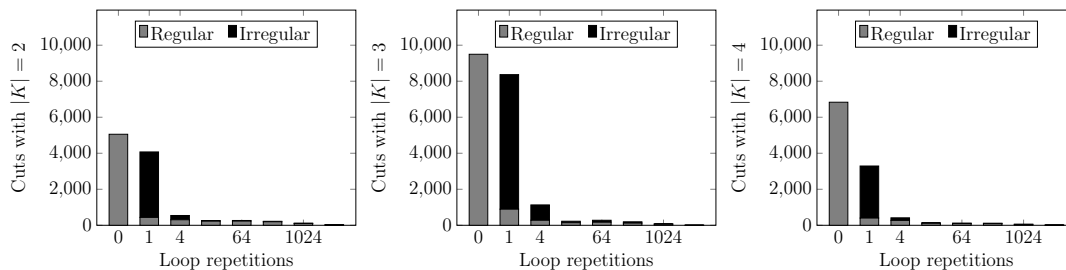

13

Figure 3: Incidence of regular and irregular cuts from disjunctions with |K| = 3 and |K| = 4 according to the incidence of irregular cuts across all disjunctions from 2-subsets of K, i.e., all K' ⊂ K such that |K'|=2.
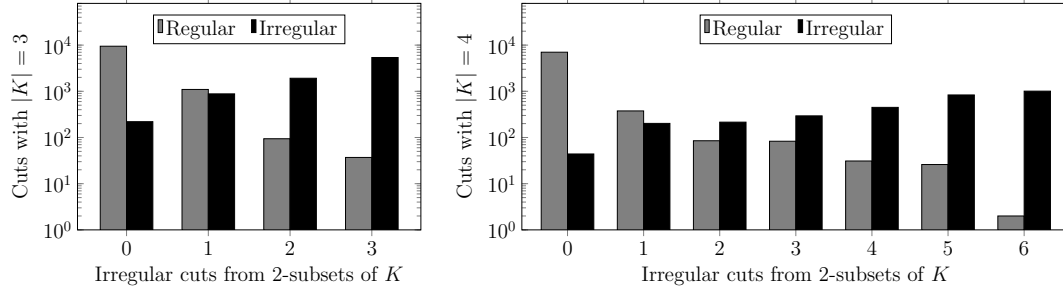

Figure 4: Comparison of regular vs. irregular cuts on instances and sizes of K where both types are found: the average gap closed is larger in 22 cases for irregular cuts vs. 24 cases for regular cuts; and the average Euclidean distance of the cut is larger in 37 cases for irregular cuts vs. 34 for regular cuts.

14
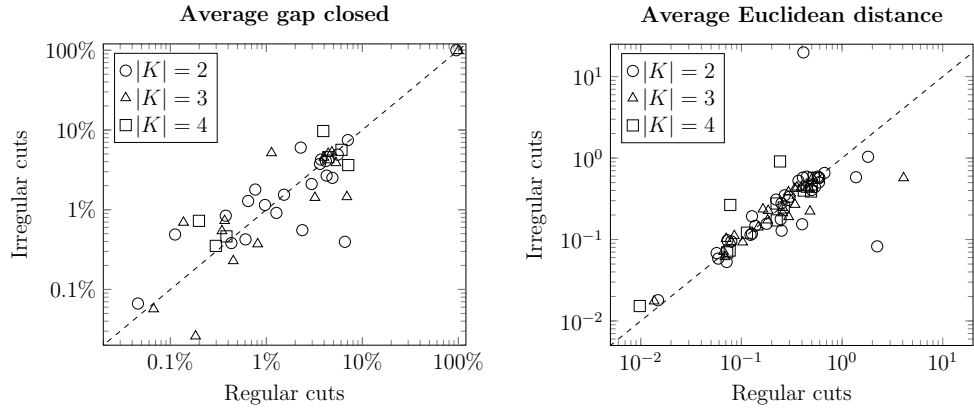
Table 1: Results for instances where lift-and-project cuts with |K|=2 and |K|=3 are generated and tested. For each instance and size of K, we report the regularity of the CGLP solution, when those solutions define split cuts, and the regularity of the cut by running Algorithm 1 with time limit of one hour.

Instance	Fractional Variables	Lift-and-project cuts with $|K| = 2$						Lift-and-project cuts with $|K| = 3$					
		CGLP basis		Cut				CGLP basis		Cut			
		Regular	Irregular	Split	Regular	Irregular	Unknown	Regular	Irregular	Split	Regular	Irregular	Unknown
air01	5	4	6	0	4	6	0	1	9	0	1	5	4
bell5	25	240	60	64	280	9	11	1787	513	82	2090	63	147
blend2	6	4	11	2	9	5	1	8	12	1	15	4	1
bm23	6	0	15	0	0	15	0	0	20	0	0	20	0
enigma	8	4	24	2	25	0	3	4	52	3	45	0	11
flugpl	10	16	29	5	16	29	0	33	87	1	35	85	0
gt2	11	55	0	0	55	0	0	153	12	0	164	0	1
khb05250	19	10	161	10	11	160	0	0	969	0	6	957	6
lseu	11	43	12	31	51	4	0	92	73	46	150	15	0
markshare1	6	0	15	0	0	15	0	0	20	0	0	20	0
markshare2	7	10	11	10	10	11	0	9	26	9	10	25	0
mas74	12	65	1	61	66	0	0	203	17	195	220	0	0
mas76	11	55	0	55	55	0	0	165	0	165	165	0	0
misc01	12	64	2	47	64	0	2	166	54	66	175	13	32
misc02	8	26	2	19	26	2	0	32	24	15	32	14	10
misc03	14	87	4	76	87	1	3	334	30	256	334	0	30
misc05	11	17	38	14	21	22	12	18	147	6	27	64	74
misc07	16	115	5	108	115	0	5	502	58	442	502	0	58
mod008	5	0	10	0	4	6	0	0	10	0	3	7	0
mod013	5	4	6	4	6	4	0	2	8	1	6	3	1
modglob	30	2	433	1	32	403	0	0	4060	0	75	3983	2
p0033	6	13	2	11	15	0	0	19	1	13	20	0	0
p0040	4	5	1	2	6	0	0	3	1	0	4	0	0
p0201	20	189	1	144	190	0	0	1124	16	448	1140	0	0
p0282	26	241	84	185	255	48	22	1585	1015	983	1703	471	426
p0291	10	30	15	16	34	7	4	70	50	13	76	26	18
pipex	6	3	12	0	8	7	0	0	20	0	8	7	5
pk1	15	1	104	1	9	96	0	0	455	0	4	451	0
rgn	14	40	51	39	65	25	1	76	288	70	190	152	22
sample2	12	18	48	7	21	44	1	28	192	0	31	175	14
sentoy	8	1	27	1	11	17	0	0	56	0	28	28	0
stein9	6	9	6	1	13	2	0	2	18	0	12	8	0
stein15	12	9	57	1	13	53	0	0	220	0	0	220	0
stein27	21	42	168	15	45	164	1	8	1322	1	14	1316	0
vpm1	15	100	5	98	102	3	0	441	14	436	450	5	0
vpm2	30	314	121	311	390	38	7	2632	1428	2615	3441	461	158
Instance average		51.3%	48.7%	34.6%	62.9%	35.0%	2.0%	40.8%	59.2%	22.5%	54.5%	38.5%	7.0%

Figure 5: Comparison of total gap closed with and without irregular cuts on instances and sizes of K where both types are found: the total gap closed with irregular cuts is larger in 27 cases.
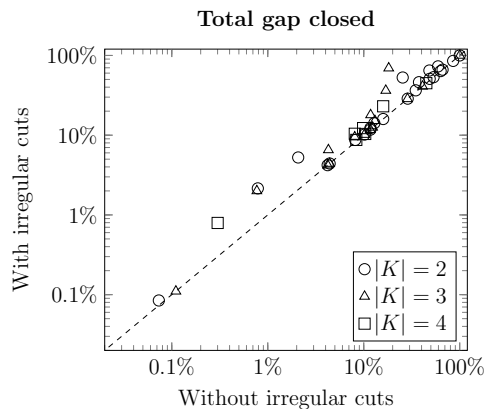
For the remaining 29 instances reported in Table 5, we repeat the loop in Algorithm 1 once. This set includes some instances previously excluded.

7   Discussion

The experiments above evidence some trends on irregular CGLP bases and strictly irregular cuts, from which we can draw observations about their incidence and relation to split cuts.

First, there is a substantial difference between the number of irregular CGLP bases and that of strictly irregular cuts. In all tables, the per-instance averages differ in at least 10% with respect to the total number of cuts. Therefore, an irregular CGLP basis often does not guarantee that the corresponding cut is strictly irregular.

Second, most regular and irregular cuts can nevertheless be identified with the first test applied on them. Such paradox is in part explained by the fact that strictly irregular cuts are less frequent than regular cuts in our experiments. Figure 2 shows that most regular cuts are identified immediately because they come from regular CGLP solutions, whereas most irregular cuts are identified in the first repetition of the loop of Algorithm 1. The same is true for the larger problems reported in Table 5: with a few exceptions, the majority of the cuts is identified as regular or irregular even though the loop is never repeated. In other words, restricting the number of nonzero multipliers of the linear relaxation to n across all terms of the CGLP formulation is an effective way of determining if a cut is irregular. Conversely, most strictly irregular cuts are those that can only be derived by using more than n rows of the linear relaxation. Hence, the linear dependence among at most n rows is a secondary factor for cut irregularity in practice.

Table 2: Additional results for instances where lift-and-project cuts with |K|=4 are also generated and tested.

| | | Lift-and-project cuts with $|K| = 4$ | | | | | |
| | | CGLP basis | | Cut | | | |
Instance	Fractional Variables	Regular	Irregular	Split	Regular	Irregular	Unknown
blend2	6	4	11	0	6	9	0
bm23	6	0	15	0	0	15	0
enigma	8	4	66	3	68	1	1
flugpl	10	52	158	0	71	139	0
gt2	11	179	151	0	238	1	91
markshare1	6	0	15	0	0	15	0
markshare2	7	1	34	1	1	34	0
misc03	14	850	151	508	856	0	145
mod008	5	0	5	0	1	4	0
mod013	5	2	3	0	3	2	0
p0033	6	15	0	8	15	0	0
p0040	4	0	1	0	1	0	0
p0201	20	4191	654	560	4753	0	92
p0291	10	106	104	6	109	34	67
pipex	6	0	15	0	3	5	7
pk1	15	0	1365	0	0	1365	0
rgn	14	74	927	62	317	539	145
sample2	12	17	478	0	21	430	44
sentoy	8	0	70	0	31	39	0
stein9	6	0	15	0	12	3	0
stein15	12	0	495	0	0	495	0
vpm1	15	1339	26	1336	1347	17	1
Instance average		26.6%	73.4%	5.2%	47.3%	46%	6.7%
Average with $|K| = 3$		35.1%	64.9%	15.8%	53.3%	42.3%	4.4%
Average with $|K| = 2$		43.7%	56.3%	26.4%	59.6%	38.9%	1.5%

Table 3: Frequency of irregular bases and cuts with |K|=2, 3, and 4.

| | Irregularity with $|K| = 2$ | | Irregularity with $|K| = 3$ | | Irregularity with $|K| = 4$ | |
Instance	Basis	Cut	Basis	Cut	Basis	Cut
blend2	73.3%	33.3%	60%	20%	73.3%	60%
bm23	100%	100%	100%	100%	100%	100%
enigma	85.7%	0%	92.9%	0%	94.3%	1.4%
flugpl	64.4%	64.4%	72.5%	70.8%	75.2%	66.2%
gt2	0%	0%	7.3%	0%	45.8%	0.3%
markshare1	100%	100%	100%	100%	100%	100%
markshare2	52.4%	52.4%	74.3%	71.4%	97.1%	97.1%
misc03	4.4%	1.1%	8.2%	0%	15.1%	0%
mod008	100%	60%	100%	70%	100%	80%
mod013	60%	40%	80%	30%	60%	40%
p0033	13.3%	0%	5%	0%	0%	0%
p0040	16.7%	0%	25%	0%	100%	0%
p0201	0.5%	0%	1.4%	0%	13.5%	0%
p0291	33.3%	15.6%	41.7%	21.7%	49.5%	16.2%
pipex	80%	46.7%	100%	35%	100%	33.3%
pk1	99%	91.4%	100%	99.1%	100%	100%
rgn	56%	27.5%	79.1%	41.8%	92.6%	53.8%
sample2	72.7%	66.7%	87.3%	79.5%	96.6%	86.9%
sentoy	96.4%	60.7%	100%	50%	100%	55.7%
stein9	40%	13.3%	90%	40%	100%	20%
stein15	86.4%	80.3%	100%	100%	100%	100%
vpm1	4.8%	2.9%	3.1%	1.1%	1.9%	1.2%
Instance average	56.3%	38.9%	64.9%	42.3%	73.4%	46%

Table 4: Results for instances where only lift-and-project cuts with $|K| = 2$ are generated and tested.

| | | Lift-and-project cuts with $|K| = 2$ | | | | | |
| | Fractional | CGLP basis | | Cut | | | |
Instance	Variables	Regular	Irregular	Split	Regular	Irregular	Unknown
bell3a	32	404	92	26	458	12	26
bell3b	36	442	188	124	568	33	29
bell4	46	887	148	289	966	18	51
dcmulti	49	503	673	439	526	631	19
egout	40	16	764	0	435	320	25
noswot	22	189	42	90	199	0	32
p0548	48	313	815	145	648	138	342
pp08a	53	35	1343	0	35	1343	0
pp08aCUTS	46	430	605	0	636	295	104
Instance average		48.4%	51.6%	15.8%	64.9%	26.9%	8.2%

18

Table 5: Remaining results for larger instances where lift-and-project cuts with $|K| = 2$ are generated and tested, but the loop of Algorithm 1 is only repeated once.

| | | Lift-and-project cuts with $|K| = 2$ | | | | | |
| | Fractional | CGLP basis | | Cut | | | |
Instance	Variables	Regular	Irregular	Split	Regular	Irregular	Unknown
aflow30a	31	155	310	130	157	296	12
aflow40b	38	347	356	337	349	240	114
cracpb1	40	389	391	364	408	259	113
dsbmip	48	464	664	453	871	36	221
fiber	42	687	174	657	713	42	106
fixnet3	69	198	2148	0	228	0	2118
fixnet4	67	180	2031	0	196	0	2015
fixnet6	60	139	1631	0	153	0	1617
gen	41	505	315	483	511	58	251
gesa2	58	704	949	655	749	7	897
gesa2_o	73	1356	1272	1251	1383	763	482
gesa3	85	1785	1785	1717	1820	589	1161
gesa3_o	100	2175	2775	2092	2266	1555	1129
glass4	72	2556	0	2556	2556	0	0
harp2	30	435	0	0	435	0	0
l152lav	51	254	1021	228	292	904	79
lp4l	23	50	203	27	73	121	59
opt1217	27	74	277	48	88	106	157
p2756	77	1521	1405	452	2190	282	454
qiu	36	474	156	404	474	16	140
qnet1	47	578	503	564	645	12	424
qnet1_o	11	20	35	19	32	0	23
rout	35	304	291	298	308	220	67
set1al	218	239	23414	19	276	20940	2437
set1ch	138	122	9331	30	150	8786	517
set1cl	220	180	23910	19	225	21393	2472
timtab1	134	2829	6082	2360	2942	5935	34
timtab2	236	8929	18801	6856	9172	18027	531
tr12-30	348	47	60331	0	605	59046	727
Instance average		38.7%	61.3%	30.5%	43.0%	31.8%	25.2%

Third, simple t-branch split disjunctions with more terms yield more irregular cuts and even more irregular CGLP bases. We can observe that first from the difference in per-instance averages of the corresponding columns for $|K| = 2$ and $|K| = 3$ in Table 1 and from $|K| = 2$ to $|K| = 4$ in Table 3. In addition, the frequency of irregular CGLP bases and cuts for each instance often increases with the size of K, the former more than the latter. For the 36 instance reported in Table 1, the frequency of irregular CGLP bases for $|K| = 3$ is higher in 29 instances and lower in 3 when compared to the case of $|K| = 2$, whereas the frequency of irregular cuts is higher in 17 and lower in 9. For the 22 instances reported in Table 3, there are proportionally more irregular CGLP bases for $|K| = 4$ in 15

instances and less in 2 instances when compared to $|K| = 2$, while the ratio of irregular cuts is higher in 12 instances and lower in 4 instances. Similarly, disjunction with more terms yield proportonially less basic regular CGLP solutions mapping to split cuts. Hence, a disjunction with more terms seems more likely to yield lift-and-project cuts that are not obtainable from a split disjunction.

Fourth, some disjunctions are more likely to yield irregular cuts than others. For simple t-branch split disjunctions, we observe in Figure 3 that an irregular cut obtained from a disjunction defined by a set K is such that the disjunctions defined by subsets of K of size 2 often yield irregular cuts as well. Conversely, if we know which sets of size 2 define disjunctions yielding irregular cuts, then a union of such sets defines a disjunction on more terms that is more likely to yield irregular cuts. For example, a disjunction on variables x_A, x_B, and x_C yields an irregular cut with increasing probability as more irregular cuts are obtained among disjunctions on x_A and x_B, x_A and x_C, and x_B and x_C. The probability of obtaining an irregular cut from a disjunction defined by a set K of size 3 or 4 is already close to 50% if at least one subset of K of size 2 defines a disjunction yielding an irregular cut. A simpler hypothesis that could be ventured is that the presence of certain variables in the disjunction makes it more likely that the CGLP would yield irregular cuts, although it is yet to be determined what makes disjunctions on particular variables more prone to yield irregular cuts. Nevertheless, our results indicate that one could use information about cuts from disjunctions where $|K| = 2$ to augment and combine those sets of indices yielding irregular cuts in the hope of obtaining irregular cuts on disjunctions with more terms.

Fifth, the frequency of irregular cuts depends on the structure of the problem, which can be higher in larger instances but lower when more inequalities are added. This comes from observing the results for the many families of instances in the experiments. For families with similar size, we observe that some have little irregularity in the results (`mas74` and `mas76`, all instances prefixed with `misc` except `misc05`, instances prefixed with `bell`, and instances prefixed with `fixnet`), some have a moderate level (`markshare1` and `markshare2`, `mod008` and `mod013`, and `aflow30a` and `aflow40b`), and some are highly irregular (instances prefixed with `set1` and instances prefixed with `timtab`). Notably, instances in the latter sets are often larger. We also observe instances yielding more irregular cuts according to their size among instances prefixed with `p` (`p0033` to `p0291`), `stein9` to `stein27`, and `vpm1` and `vpm2`. Curiously, however, we observe a reduction in the number of strictly irregular cuts when comparing pairs of instances that represent a same problem. For example, instances `pp08a`, `gesa2_o`, `gesa3_o`, and `qnet1_o` are respectively equivalent to instances `pp08aCUTS`, `gesa2`, `gesa3`, and `qnet1`. However, the latter four have additional constraints strengthening the formulation. Among those, the number of strictly irregular cuts only increases from `qnet1_o` to `qnet1`, but in that case the number of fractional variables is substantially smaller and that may have affected the results. This finding resonates with the example in Figure 1d, which shows that additional inequalities can make more of the valid cuts regular. We could hypothesize that these additional constraints extend the reach of regular cuts, hence making irregular cuts less relevant than before. Overall, these results

indicate that cuts from disjunctions with more terms would yield more irregular cuts at the root node and that they are more likely to do so in larger problems.

Finally, irregular cuts are as strong as regular cuts, and thus could complement them if necessary. When comparing the average gap closed or the average Euclidean distance to the fractional solution separated, there are about the same number of cases favoring either type of cut. Figure 5 shows that, when both regular and irregular cuts are generated for a given problem, part of the optimality gap closed in many of the instances is due exclusively to the irregular cuts. In fact, some irregular cuts may be obtained as regular cuts of higher rank. For example, the irregular cut in Figure 1c can be obtained as a regular cut of the tightened LP relaxation in Figure 1b, but the use of previously generated cuts in the LP tableau to generate further intersection cuts may propagate numerical errors from those previous cuts as well. Hence, directly generating those cuts may be preferable in certain cases. From Figure 4, we can note that the results for the average Euclidean distance fluctuate at a much smaller scale around the identity line, possibly indicating that outliers to either side could be a starting point to refine the analysis in future work.

Most of the related computational work has focused on regular cuts, including Balas and Qualizza (2013), which is understandable since those are the cuts that we know how to generate more easily. Even among those, Louveaux et al. (2015) note that general-purpose cut generation is not yet competitive with specific methods, and that using too many rows of the tableau of the LP relaxation is not necessarily better. In fact, most work is based on methods that are tailored for the different types of maximal lattice-free sets used to generate intersection cuts (Espinoza, 2008; Basu et al., 2011; Balas and Qualizza, 2013; Dey et al., 2014). Future work on irregular cuts should probably focus on tailored methods.

One exception to the focus on regular cuts is the work on cross-cuts by Dash et al. (2014), which also formulates a CGLP based on non-split disjunctions. Interestingly, some ideas in this paper resonate with our observations and findings. One of their best approaches is denoted as Cross.def, which generates cross cuts that could only be otherwise obtained from split disjunctions as cuts of rank 2. Similarly, we have argued that irregular cuts can be regular cuts of higher rank, even for the same type of disjunction. In another approach described in the paper, one of the disjunctions is fixed and a second disjunction is chosen to generate a cut through an MILP formulation. Our finding about how simple 3-branch and 4-branch split disjunctions yielding strictly irregular cuts are associated with those yielding strictly irregular cuts among simple 2-branch split disjunctions paves the way to judiciously augment disjunctions when aiming to generate strictly irregular cuts.

8   Conclusion

In this paper we have used a mixed-integer formulation to determine when the equivalence between lift-and-project cuts from arbitrary disjunctions and intersection cuts does not hold. This method is conveniently used to evaluate the extent to which unstrengthened

t-branch split cuts differ from multi-row cuts, two types of cuts that have been intensely studied for the past decade. When there is no equivalence, the cut is said to be irregular and can only be obtained from irregular CGLP solutions.

We have focused our analysis on strictly irregular cuts, which we have shown to conveniently exclude irregular cuts that are implied by the set of regular cuts. On the one hand, we have found that the incidence of irregular cuts varies across different families of instances and that many irregular CGLP solutions nevertheless correspond to regular cuts. On the other hand, the incidence of irregular cuts often increases as the problems get larger, the linear relaxation is weaker, and the disjunction has more terms. We also found that proportionally less basic regular CGLP solutions characterize split cuts in disjunctions with more terms. Interestingly, we observed in our experiments that irregular cuts are on average as strong as regular cuts and that they are often responsible for part of the optimality gap closed when both types of cuts are generated. Hence, irregular cuts could complement their regular counterparts, in particular to avoid the numerical instability of higher rank cuts that could otherwise be necessary to solve some problems. In such case, the only irregular cuts that would be relevant are precisely the strictly irregular ones that we analyzed in this study. Furthermore, we have observed that 3-branch and 4-branch split disjunctions yield irregular cuts more often when they augment multiple 2-branch split disjunctions that also yield irregular cuts, a result that could be used to judiciously choose disjunctions with more terms from which to generate additional cuts. Ultimately, given that we would be mostly interested in irregular cuts when generating lift-and-project cuts from arbitrary disjunctions, knowing which disjunctions are more likely to generate irregular cuts can be extremely helpful.

References

T. Achterberg, T. Koch, and A. Martin. MIPLIB 2003. *Operations Research Letters*, 34 (4):361–372, 2006.

K. Andersen, G. Cornuéjols, and Y. Li. Split closure and intersection cuts. *Mathematical Programming*, 102(3):457–493, 2005.

K. Andersen, Q. Louveaux, R. Weismantel, and L. A. Wolsey. Inequalities from two rows of a simplex tableau. In M. Fischetti and D. P. Williamson, editors, *Integer Programming and Combinatorial Optimization*, pages 1–15, 2007.

E. Balas. Intersection cuts – a new type of cutting planes for integer programming. *Oper. Res.*, 19:19–39, 1971.

E. Balas. Disjunctive programming. *Annals of Discrete Mathematics*, (5):3–51, 1979.

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

E. Balas and P. Bonami. Generating lift-and-project cuts from the LP simplex tableau: open source implementation and testing of new variants. *Mathematical Programming Computation*, 1(2-3):165–199, 2009.

E. Balas and R. G. Jeroslow. Strengthening cuts for mixed integer programs. *European Journal of Operational Research*, 4(4):224 – 234, 1980.

E. Balas and T. Kis. On the relationship between standard intersection cuts, lift-and-project cuts, and generalized intersection cuts. *Math. Program.*, 160:85–114, 2016.

E. Balas and M. Perregaard. A precise correspondence between lift-and-project cuts, simple disjunctive cuts, and mixed integer gomory cuts for 0–1 programming. *Math. Program., Ser. B*, 94:221–245, 2003.

E. Balas and A. Qualizza. Intersection cuts from multiple rows: A disjunctive programming approach. *EURO J. Comput. Optim.*, 1:3–49, 2013.

E. Balas, S. Ceria, and G. Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0–1 programs. *Math. Program.*, 58:295–324, 1993.

E. Balas, S. Ceria, and G. Cornuéjols. Mixed 0–1 programming by lift-and-project in a branch-and-cut framework. *Manage. Sci.*, 42:1229–1246, 1996.

A. Basu, M. Conforti, G. Cornuéjols, and G. Zambelli. Maximal lattice-free convex sets in linear subspaces. *Mathematics of Operations Research*, 35(3):704–720, 2010.

A. Basu, P. Bonami, G. Cornuéjols, and F. Margot. Experiments with two-row cuts from degenerate tableaux. *INFORMS Journal on Computing*, 23(4):578–590, 2011.

A. Basu, M. Campêlo, M. Conforti, G. Cornuéjols, and G. Zambelli. Unique lifting of integer variables in minimal inequalities. *Mathematical Programming*, 141(1):561–576, 2013.

A. Basu, M. Conforti, and M. Di Summa. A geometric approach to cut-generating functions. *Mathematical Programming*, 151(1):153–189, 2015.

R. E. Bixby, E. A. Boyd, and R. R. Indovina. MIPLIB: A test set of mixed integer programming problems. *SIAM News*, 25:16, 1992.

R. E. Bixby, S. Ceria, C. M. McZeal, and M. W. P Savelsbergh. An updated mixed integer programming library: MIPLIB 3.0. *Optima*, (58):12–15, 1998.

V. Borozan and G. Cornuéjols. Minimal valid inequalities for integer constraints. *Mathematics of Operations Research*, 34(3):538–546, 2009.

M. Conforti, G. Cornuéjols, and G. Zambelli. A geometric perspective on lifting. *Operations Research*, 59(3):569–577, 2011a.

M. Conforti, G. Cornuéjols, and G. Zambelli. Corner polyhedron and intersection cuts. *Surveys in Operations Research and Management Science*, 16(2):105 – 120, 2011b.

G. Cornuéjols and F. Margot. On the facets of mixed integer programs with two integer variables and two constraints. *Mathematical Programming*, 120(2):429–456, 2008.

S. Dash, S. S. Dey, and O. Günlük. Two dimensional lattice-free cuts and asymmetric disjunctions for mixed-integer polyhedra. *Mathematical Programming*, 135(1):221–254, 2012.

S. Dash, O. Günlük, and J. P. Vielma. Computational experiments with cross and crooked cross cuts. *INFORMS Journal on Computing*, 26(4):780–797, 2014.

S. S. Dey and L. A. Wolsey. Two row mixed-integer cuts via lifting. *Mathematical Programming*, 124(1):143–174, 2010.

S. S. Dey, A. Lodi, A. Tramontani, and L. A. Wolsey. On the practical strength of two-row tableau cuts. *INFORMS Journal on Computing*, 26(2):222–237, 2014.

D. G. Espinoza. Computing with multi-row gomory cuts. In A. Lodi, A. Panconesi, and G. Rinaldi, editors, *Integer Programming and Combinatorial Optimization*, pages 214–224. Springer Berlin Heidelberg, 2008.

R. Fukasawa, L. Poirrier, and Á. S. Xavier. The (not so) trivial lifting in two dimensions. Technical report, University of Waterloo, 2016.

R. E. Gomory. An algorithm for the mixed integer problem. Technical report, The Rand Corporation, 1960.

T. Kis. Lift-and-project for general two-term disjunctions. *Discrete Optimization*, 12:98 – 114, 2014.

Y. Li and J.-P. P. Richard. Cook, Kannan and Schrijver's example revisited. *Discrete Optimization*, 5(4):724 – 734, 2008.

Q. Louveaux, L. Poirrier, and D. Salvagnin. The strength of multi-row models. *Mathematical Programming Computation*, 7(2):113–148, 2015.