Experiments with two-row cuts from degenerate tableaux

Amitabh Basu
Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA 15213.
Pierre Bonami
LIF, Faculté des Sciences de Luminy, Université de Marseille, France.
Gérard Cornuéjols
Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA 15213; and LIF, Faculté des Sciences de Luminy, Université de Marseille, France.
François Margot
Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA 15213.

There has been a recent interest in cutting planes generated from two or more rows of the optimal simplex tableau. One can construct examples of integer programs for which a single cutting plane generated from two rows dominates the entire split closure. Motivated by these theoretical results, we study the effect of adding a family of cutting planes generated from two rows on a set of instances from the MIPLIB library. The conclusion of whether these cuts are competitive with Gomory Mixed Integer cuts is very sensitive to the experimental setup. In particular, we consider the issue of reliability versus aggressiveness of the cut generators, an issue that is usually not addressed in the literature.

Key words: Mixed Integer Programming, Cutting Planes.

1 Introduction

In the last 15 years, generic cutting planes have played a major role in the progress of mixed integer linear programming (MILP) solvers. Most cutting plane algorithms available in state of the art solvers rely on cuts that can be derived from a single equation (note that the equation used to derive cuts need not be one of the constraints of the problem; it may be obtained as a linear combination of the constraints). Examples are Gomory Mixed Integer cuts (Gomory, 1963), MIR cuts (Marchand and Wolsey, 2001), lift-and-project cuts (Balas et al., 1993), or lifted cover inequalities (Crowder et al., 1983). A natural extension is to derive cuts from more than one equation, an area of research enjoying a revival in recent years.

The study of the Corner Polyhedron started by Gomory and Johnson (Gomory, 1969; Gomory and Johnson, 1972; Johnson, 1974), and of Intersection Cuts introduced by Balas (1971) provides a framework for generating cutting planes using multiple rows of the simplex tableau. Several papers give theoretical results on cuts that can be derived from two rows: Andersen et al. (2007); Basu et al. (2009); Borozan and Cornuéjols (2009); Cornuéjols and Margot (2009); Dey and Wolsey (2008). In Basu et al. (2009), we provide examples of integer programs with two constraints and two integer variables, where the integrality gap can be closed by a single inequality derived from the two constraints, while the value of the linear relaxation obtained by adding all split cuts is arbitrarily close to 0. These examples suggest that cuts derived from two rows might improve the efficiency of MILP solvers. In this paper, we test the empirical effectiveness of a specific class of two-row cuts inspired by these examples.

The paper is organized as follows. In Section 2, we recall basic results on two-row cuts and motivate the choice of the triangles used in our separation algorithm. In Section 3, we give closed form formulas for certain types of cuts, discuss integer lifting of these cuts, as well as their strengthening when the basic variables are nonnegative. In Section 4, we present computational results on a standard set of test problems. We show that the conclusions are very sensitive to the experimental setup. We present two different setups which lead to opposite conclusions and argue that one of them is flawed because it overlooks the issues of reliability versus aggressiveness of the cut generator. Our conclusion is that the class of two-row cuts investigated in this paper only improves marginally over classical Gomory Mixed Integer cuts.

2 Two-row cutting planes

2.1 Basic theory

In this paper, we study a class of cutting planes derived from two equations. Consider a mixed-integer problem with two free integer variables and a finite number of nonnegative, continuous variables.

$$\begin{array}{rcl} x &=& f + \sum_{j=1}^{k} r^j s_j \\ x &\in& \mathbb{Z}^2 \\ s &\in& \mathbb{R}^k_+ \; . \end{array} \tag{1}$$

We assume $f \in \mathbb{Q}^2 \setminus \mathbb{Z}^2$, $k \geq 1$, and for $j = 1, \ldots, k$, $r^j \in \mathbb{Q}^2 \setminus \{0\}$.

Model (1) naturally arises as a relaxation of a MILP having a basic feasible solution of its linear relaxation with at least two basic integer variables, at least one of which takes a fractional value in the optimal solution. Indeed, consider the simplex tableau corresponding to the basic solution and keep only two equations corresponding to these two basic integer variables. Model (1) was introduced by Andersen et al. (2007). Several recent papers discuss some of its theoretical properties such as characterization of facets and relative strength of classes of facets (Borozan and Cornuéjols, 2009; Basu et al., 2009; Cornuéjols and Margot, 2009; Dey and Wolsey, 2008) but little empirical evidence on the strength of the resulting cuts is available (Espinoza, 2008; Dey et al., 2010). The remainder of this section is a short summary of results obtained in Borozan and Cornuéjols (2009); Basu et al. (2009); Cornuéjols and Margot (2009); Dey and Wolsey (2008) that are directly relevant to the work in this paper.

The inequalities $s_j \geq 0$ are called *trivial* for (1). A nontrivial valid inequality for (1) is of the form

$$\sum_{j=1}^{k} \psi(r^j) s_j \geq 1 \ , \tag{2}$$

where $\psi : \mathbb{R}^2 \to \mathbb{R}_+$. A nontrivial valid inequality is *minimal* if there is no other nontrivial valid inequality $\sum_{j=1}^{k} \psi'(r^j) s_j \geq 1$ such that $\psi'(r^j) \leq \psi(r^j)$ for all $j = 1, \ldots, k$. The nontrivial inequalities can be seen as a particular case of intersection cuts (Balas, 1971). Minimal nontrivial valid inequalities are associated with functions ψ that are nonnegative positively-homogeneous piecewise-linear and convex. Furthermore, the set

$$B_\psi := \{x \in \mathbb{R}^2 : \ \psi(x - f) \leq 1\} \tag{3}$$

is a maximal lattice-free convex set with f in its interior (Borozan and Cornuéjols, 2009). By *lattice-free convex set* we mean a convex set with no integral point in its interior. However integral points are allowed on the boundary. These maximal lattice-free convex sets are splits, triangles, and quadrilaterals as proved by Lovász (1989). In this paper, we will focus on splits and triangles only.

A *maximal split* is a set of the form $c \leq ax_1 + bx_2 \leq c + 1$ where a and b are coprime integers and c is an integer.

Following Dey and Wolsey (2008), we partition the maximal lattice-free triangles into three types (see Figure 1):

- *Type 1 triangles*: triangles with integral vertices and exactly one integral point in the relative interior of each edge;

- *Type 2 triangles*: triangles with at least one fractional vertex v, exactly one integral point in the relative interior of the two edges incident to v and at least two integral points on the third edge;
- *Type 3 triangles*: triangles with exactly three integral points on the boundary, one in the relative interior of each edge.

Figure 1 shows these three types of triangles as well as a maximal lattice-free quadrilateral and a split.

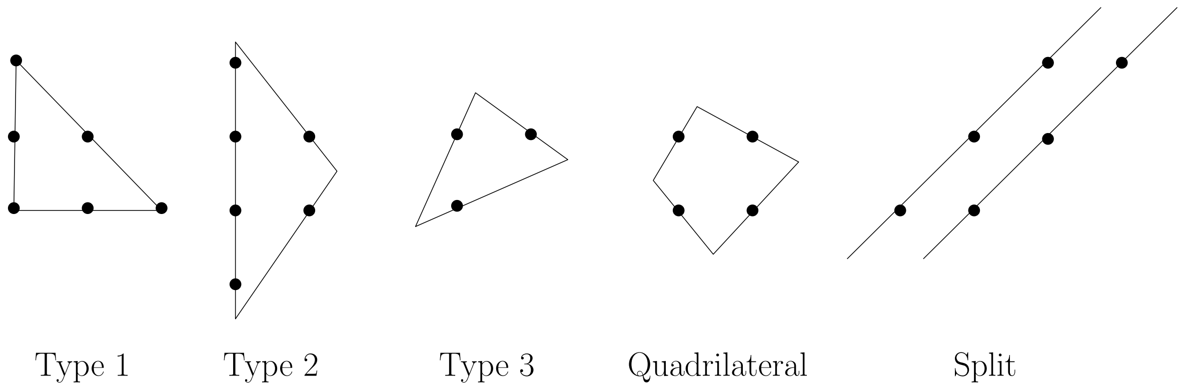

Figure 1: Maximal lattice-free convex sets with nonempty interior in $\mathbb{R}^2$

In this paper we focus on simple classes of split and Type 2 triangles. We do not insist on generating facets of the convex hull of (1). We motivate our choice of cuts in Section 2.2. In Section 3, we give closed form formulas for our cuts.

2.2 Type 2 triangles versus splits

For model (1), the *split closure* is defined as the intersection of all minimal valid inequalities obtained from splits (Cook et al., 1990). Similarly, the *Type 2 triangle closure* is the intersection of all the minimal valid inequalities generated from Type 2 triangles. The Type 2 triangle closure approximates the convex hull of (1) within a factor of 2 (see Basu et al., 2009, Theorem 7.3 for a precise statement). On the other hand, the split closure is not always a good approximation of the convex hull of (1). In Basu et al. (2009), we construct integer programs where a single inequality derived from a Type 2 triangle closes the integrality gap entirely whereas the split closure closes an arbitrarily small fraction of the gap. Specifically, let η and μ be large positive numbers and $0 < \nu < 1$. We choose $f = (0, \nu)$ and a lattice-free triangle T of Type 2 with vertices p^1, p^2, p^3 where $p^2 = (-1, 1 + \eta)$, $p^3 = (-1, -\mu)$ and p^1 is the intersection point of the lines going through p^3 and $(0, 0)$, and p^2 and $(0, 1)$. The integer program is

$$\begin{array}{rcl} \min & & s_1 + s_2 + s_3 \\ x & = & f + r^1 s_1 + r^2 s_2 + r^3 s_3 \\ x & \in & \mathbb{Z}^2 \\ s & \in & \mathbb{R}^3_+ \ , \end{array} \tag{4}$$

where r^1, r^2 and r^3 are the rays defined by $r^j = p^j - f$, $j = 1, 2, 3$. The intersection cut associated with T is $s_1 + s_2 + s_3 \geq 1$ and closes the integrality gap entirely. On the other hand, the split closure only closes a small fraction of the integrality gap when $\eta + \mu$ is large. See the above reference for details.

This example suggests that it might be interesting to generate cuts from Type 2 triangles in addition to the classical Gomory Mixed Integer cuts (Nemhauser and Wolsey, 1990) when solving an MILP. In this paper, we separate a very restricted class of Type 2 triangle cuts. This class is designed so that separation and integer lifting can be done through simple closed form formulas. In addition, some of the bounds on the basic variables can also easily be used to obtain alternative cuts. Consider the simplex tableau obtained by solving the LP relaxation of MILP. We have

$$x = f + \sum_{j=1}^{k} r^j s_j$$

where $x \in \mathbb{R}^m$ denotes the basic variables and $s \in \mathbb{R}^k$ the nonbasic. Motivated by the example mentioned above, we test the effectiveness of cutting planes from two rows of the simplex tableau

$$\begin{array}{rcl} x_i & = & f_i + \sum_{j=1}^{k} r_i^j s_j \\ x_\ell & = & f_\ell + \sum_{j=1}^{k} r_\ell^j s_j \end{array} \tag{5}$$

where x_i and x_ℓ are basic integer variables, f_i is integral and f_ℓ is fractional, i.e. at least one of the basic integral variables has an integral value and at least one has a fractional value. Note that this will typically occur when the basis is degenerate. Without loss of generality we can make a translation so that $f_i = 0$ and $f_\ell \in]0, 1[$. We try to construct two Type 2 triangles, one where the edge that contains at least two integral points is on the line $x_i = -1$, and a symmetrical one where this edge is on the line $x_i = 1$. As these two cases are similar, we focus on the first one in the following discussion. For each ray r^j such that the half-line $f + \lambda r^j$ for $\lambda > 0$ intersects the hyperplane $x_i = -1$, we compute that intersection and its projection onto the (x_i, x_ℓ)-plane. Among these projections, we define $p^2 = (p_i^2, p_\ell^2)$ (resp. $p^3 = (p_i^3, p_\ell^3)$) as the point with largest (resp. smallest) x_ℓ-coordinate, i.e.,

$$p^2 = \left(-1, f_\ell - \min_j \left\{ \frac{r_\ell^j}{r_i^j} \mid r_i^j < 0 \right\}\right) \qquad \text{and} \qquad p^3 = \left(-1, f_\ell - \max_j \left\{ \frac{r_\ell^j}{r_i^j} \mid r_i^j < 0 \right\}\right) .$$

Note that p^2 and p^3 are not defined if $r_j^i \geq 0$ for all $j = 1, \dots, k$. In this case or when $p^2 = p^3$, we do not generate a cut. Otherwise, we consider three cases:

i) There are at least two integral points in the interior of the segment p^2p^3. We construct a Type 2 triangle with vertices p^1, p^2 and p^3, where p^1 is the intersection of the lines passing through p^2 and $(0, 1)$ and through p^3 and $(0, 0)$.

ii) There is exactly one integral point in the interior of the segment p^2p^3. As the construction used in point i) may result in a non-lattice-free triangle, we modify it as follows. Let $q^2 = (-1, \lceil p_\ell^2 \rceil)$, and $q^3 = (-1, \lfloor p_\ell^3 \rfloor)$. If $\lceil p_\ell^2 \rceil - p_\ell^2 \leq p_\ell^3 - \lfloor p_\ell^3 \rfloor$, we construct a Type 1 or Type 2 triangle whose vertices are p^1, q^2 and p^3 where p^1 is the intersection of the lines passing through q^2 and $(0, 1)$ and through p^3 and $(0, 0)$. Otherwise, we construct a Type 2 triangle whose vertices are p^1, p^2 and q^3, where p^1 is the intersection of the lines passing through p^2 and $(0, 1)$ and through q^3 and $(0, 0)$.

iii) There is no integral point in the interior of the segment p^2p^3. Let $q^2 = (-1, \lceil p_\ell^2 \rceil)$, and $q^3 = (-1, \lfloor p_\ell^3 \rfloor)$. We construct a split whose sides are the lines through q^2 and $(0, 1)$ and through q^3 and $(0, 0)$.

Therefore, if the optimal tableau of the LP relaxation of the MILP has m basic integer variables with an integer value and n basic integer variables that are fractional, this algorithm generates up to $2nm$ cuts from the optimal simplex tableau. Note that although the lattice free sets generated by this algorithm are maximal, the associated intersection cuts are in general not facet defining for the relaxation defined by the two rows of the tableau. We use variants of this basic algorithm for generating and testing two-row cuts.

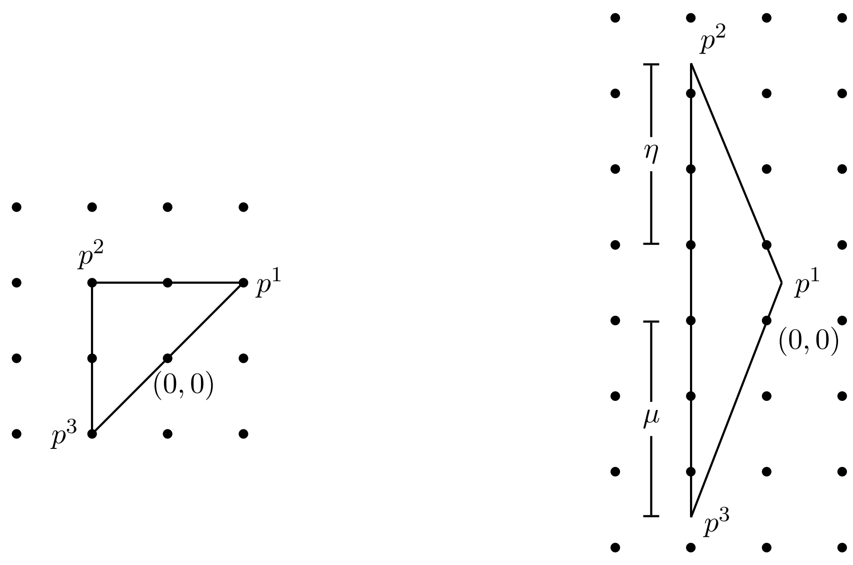

Figure 2: Type 1 triangle in canonical form ($\eta = 0$, $\mu = 1$) and Type 2 triangle in canonical form.

3 Derivation of the cuts

3.1 Formulas of the cut for Type 1 and Type 2 triangles

Consider a lattice-free triangle T of Type 1 or 2 with vertices p^1, p^2 and p^3. Without loss of generality, assume that T is in a canonical form where $p^2 = (-1, 1+\eta)$ and $p^3 = (-1, -\mu)$ with $\eta \geq 0$, $\mu > 0$. Furthermore we assume that the points $(0,0)$ and $(0,1)$ are respectively in the relative interior of edge p^3p^1 and p^2p^1. A Type 1 or Type 2 triangle can always be put into this canonical form by applying an integer translation followed by a unimodular transformation (Proposition 9 of Dey and Wolsey (2010)).

For a Type 1 triangle, we thus have $\eta = 0$ and $\mu = 1$ and the three vertices of the triangle are $p^1 = (1,1)$, $p^2 = (-1,1)$ and $p^3 = (-1,-1)$, see Figure 2. For a Type 2 triangle the points $(-1,0)$ and $(-1,1)$ are in the line segment p^2p^3, see Figure 2.

As noted in Andersen et al. (2007), the cut obtained from T is the intersection cut (Balas, 1971) derived from the disjunction

$$(-x_i \geq 1) \quad \vee \quad (\eta x_i + x_\ell \geq 1) \quad \vee \quad (\mu x_i - x_\ell \geq 0) \ . \tag{6}$$

Disjunction (6) simply states that all feasible points are not in the interior of triangle T. Using the formula for deriving the intersection cut (see Appendix A), the cut obtained from the disjunction (6) is

$$\sum_{j=1}^{k} \psi(r_i^j, r_l^j) s_j \geq 1 \tag{7}$$

with $\psi(r_i^j, r_\ell^j) = \max\left\{\frac{-r_i^j}{1+f_i}, \frac{\eta r_i^j + r_\ell^j}{1-\eta f_i - f_\ell}, \frac{\mu r_i^j - r_\ell^j}{-\mu f_i + f_\ell}\right\}$. Since T is a maximal lattice-free triangle, (7) is a minimal inequality for (1). In the next two sections, we show how the cut (7) can be strengthened if some of the non-basic variables are integer constrained, and give formulas to derive an alternative cut if the basic variable x_i is nonnegative.

3.2 Integer lifting

In this section, we consider a variation of the two-row problem (5) where some of the variables s_j are constrained to be integer. We stress that all the results in this section apply to any triangle of Type 1 or 2 in canonical form containing f in its interior. In particular, the assumption that f_i is integer is not necessary here.

$$\begin{array}{rcl} x_i & = & f_i + \sum_{j=1}^{k} r_i^j s_j \\ x_\ell & = & f_\ell + \sum_{j=1}^{k} r_\ell^j s_j \\ x_i, x_\ell & \in & \mathbb{Z}, \\ s_j & \in & \mathbb{Z} \quad \text{for all} \quad j \in I \subseteq \{1, \ldots, k\}, \\ s_j & \geq & 0 \quad \text{for all} \quad j \in \{1, \ldots, k\}. \end{array} \tag{8}$$

Clearly inequality (7) is valid for (8). Dey and Wolsey (2008) show how to strengthen the coefficients of the non-basic integer variables in (7) to obtain a minimal inequality for (8). This strengthening is called *integer lifting*. If the lattice-free set used to generate the cut is a triangle of Type 1 or 2, Dey and Wolsey (2008) show that there is a unique integer lifting that produces a minimal inequality for (8) and it is obtained through the so-called trivial fill-in function. In this section, we derive a closed-form formula for this lifting. We use the terminology of intersection cuts (Balas, 1971).

The basic idea of the trivial fill-in function is that, for each integer constrained variable s_j, the ray r^j can be translated by an integer vector before applying the function ψ to it (see Appendix B). The trivial fill-in function simply selects the integer translation which results in the smallest coefficient for s_j in the cut. To derive the formula of this strengthened cut, we define for any $j \in \{1, \dots, k\}$ and any integers m_1 and m_2:

$$\Phi_1^j(m_1) = \frac{m_1 - r_i^j}{1 + f_i},$$
$$\Phi_2^j(m_1, m_2) = \frac{\eta\left(r_i^j - m_1\right) + r_\ell^j - m_2}{1 - \eta f_i - f_\ell},$$
$$\Phi_3^j(m_1, m_2) = \frac{\mu\left(r_i^j - m_1\right) - r_\ell^j + m_2}{-\mu f_i + f_\ell},$$

and

$$\Phi^j(m_1, m_2) = \max\left\{\Phi_1^j(m_1), \Phi_2^j(m_1, m_2), \Phi_3^j(m_1, m_2)\right\}.$$

Note that $\Phi^j(m_1, m_2)$ is nothing else than $\psi(r_i^j - m_1, r_\ell^j - m_2)$. It follows directly from the properties of ψ that $\Phi^j(m_1, m_2) > 0$ for all m_1, m_2.

The strengthened cut is then given by:

$$\sum_{j \in I} \min_{(m_1, m_2) \in \mathbb{Z}^2} \Phi^j(m_1, m_2) s_j + \sum_{j \in \bar{I}} \psi(r^j) s_j \geq 1 \ ,$$

where $\bar{I} = \{1, \dots, k\} \setminus I$. The following proposition gives the closed form formula for computing $\min_{(m_1, m_2) \in \mathbb{Z}^2} \Phi^j(m_1, m_2)$.

Proposition 3.1. *Let* $\overline{m}_2^j = r_\ell^j + \frac{\left(r_i^j - \lfloor r_i^j \rfloor\right)(-\mu + (\mu + \eta) f_\ell)}{1 - (\mu + \eta) f_i}$, $\overline{m}_1^j = r_i^j + \frac{(1 + f_i)\left(r_\ell^j - \lfloor r_\ell^j \rfloor\right)}{1 + \eta - f_\ell}$ *and* $\widehat{m}_1^j = r_i^j + \frac{(1 + f_i)(\lceil r_\ell^j \rceil - r_\ell^j)}{\mu + f_\ell}$.

(i)

$$\min_{\substack{m_1, m_2 \in \mathbb{Z} \\ m_1 \leq r_i^j}} \Phi^j(m_1, m_2) = \min\left\{\Phi_2^j\left(\lfloor r_i^j \rfloor, \lfloor \overline{m}_2^j \rfloor\right), \Phi_3^j\left(\lfloor r_i^j \rfloor, \lceil \overline{m}_2^j \rceil\right)\right\};$$

(ii)

$$\min_{\substack{m_1, m_2 \in \mathbb{Z} \\ m_1 > r_i^j, m_2 \leq r_\ell^j}} \Phi^j(m_1, m_2) = \begin{cases} \min\left\{\Phi_1^j\left(\lceil \overline{m}_1^j \rceil\right), \Phi_2^j\left(\lfloor \overline{m}_1^j \rfloor, \lfloor r_\ell^j \rfloor\right)\right\} & \textit{if } \overline{m}_1^j > \lceil r_i^j \rceil, \\ \Phi_1^j\left(\lfloor r_i^j \rfloor + 1\right) & \textit{otherwise.} \end{cases};$$

(iii)

$$\min_{\substack{m_1, m_2 \in \mathbb{Z} \\ m_1 > r_i^j, m_2 \geq r_\ell^j}} \Phi^j(m_1, m_2) = \begin{cases} \min\left\{\Phi_1^j\left(\lceil \widehat{m}_1^j \rceil\right), \Phi_3^j\left(\lfloor \widehat{m}_1^j \rfloor, \lceil r_\ell^j \rceil\right)\right\} & \textit{if } \widehat{m}_1^j > \lceil r_i^j \rceil, \\ \Phi_1^j\left(\lfloor r_i^j \rfloor + 1\right) & \textit{otherwise.} \end{cases}.$$

Proof. (i) First, we note that if $m_1 \le r_i^j$ then $\Phi_1^j(m_1) = \frac{m_1 - r_i^j}{1+f_i} \le 0$. Since $\Phi^j(m_1, m_2) > 0$, this implies that $\Phi^j(m_1, m_2) = \max\left\{\Phi_2^j(m_1, m_2), \Phi_3^j(m_1, m_2)\right\}$.

As $m_1 \le r_i^j$, then $m_1 \le \lfloor r_i^j \rfloor$ and thus $r_i^j - m_1 \ge r_i^j - \lfloor r_i^j \rfloor$. Since η and μ are nonnegative, we thus have

$$\Phi_2^j(m_1, m_2) = \frac{\eta\left(r_i^j - m_1\right) + r_\ell^j - m_2}{1 - \eta f_i - f_\ell} \ge \frac{\eta\left(r_i^j - \lfloor r_i^j \rfloor\right) + r_\ell^j - m_2}{1 - \eta f_i - f_\ell} = \Phi_2^j(\lfloor r_i^j \rfloor, m_2)$$
$$\Phi_3^j(m_1, m_2) = \frac{\mu\left(r_i^j - m_1\right) - r_\ell^j + m_2}{-\mu f_i + f_\ell} \ge \frac{\mu\left(r_i^j - \lfloor r_i^j \rfloor\right) - r_\ell^j + m_2}{-\mu f_i + f_\ell} = \Phi_3^j(\lfloor r_i^j \rfloor, m_2)$$

and therefore $\min\{\Phi^j(m_1, m_2) \mid m_1, m_2 \in \mathbb{Z}, m_1 \le r_i^j\} = \min\{\Phi^j(\lfloor r_i^j \rfloor, m_2) \mid m_2 \in \mathbb{Z}\}$.

Since $\Phi_2^j(\lfloor r_i^j \rfloor, m_2)$ is decreasing in m_2, $\Phi_3^j(\lfloor r_i^j \rfloor, m_2)$ is increasing in m_2 and $\Phi_2^j(\lfloor r_i^j \rfloor, \overline{m}_2^j) = \Phi_3^j(\lfloor r_i^j \rfloor, \overline{m}_2^j)$, we have that $\min\{\Phi^j(\lfloor r_i^j \rfloor, m_2) \mid m_2 \in \mathbb{Z}\}$ is attained either in $\Phi_3^j(\lfloor r_i^j \rfloor, \lceil \overline{m}_2^j \rceil)$ or $\Phi_2^j(\lfloor r_i^j \rfloor, \lfloor \overline{m}_2^j \rfloor)$. The formula follows.

(ii) We now suppose that $m_1 > r_i^j$ and $m_2 \le r_\ell^j$. Then $\Phi_3^j(m_1, m_2) < 0$ and therefore $\Phi^j(m_1, m_2) = \max\left\{\Phi_1^j\left(m_1\right), \Phi_2^j\left(m_1, m_2\right)\right\}$. Furthermore, we have:

$$\min\{\Phi^j(m_1, m_2) \mid m_1, m_2 \in \mathbb{Z}, m_1 > r_i^j, m_2 \le r_\ell^j\} = \min\{\Phi^j(m_1, \lfloor r_\ell^j \rfloor) \mid m_1 \in \mathbb{Z}, m_1 > r_i^j\}\ .$$

To compute the minimum of this last quantity, we note that Φ_1^j is increasing in m_1, Φ_2^j is decreasing in m_1 and $\Phi_1^j(\overline{m}_1^j) = \Phi_2^j(\overline{m}_1^j, \lfloor r_l^j \rfloor)$. Therefore if $\overline{m}_1^j > \lceil r_i^j \rceil$, the minimum is attained either in $\Phi_2^j(\lfloor \overline{m}_1^j \rfloor, \lfloor r_\ell^j \rfloor)$ or $\Phi_1^j(\lceil \overline{m}_1^j \rceil, \lfloor r_\ell^j \rfloor)$, otherwise the minimum is attained in $\Phi_1^j(\lfloor r_i^j \rfloor + 1)$.

(iii) Similar to the proof of (ii). □

3.3 Using nonnegativity of basic variables

In model (1), the two basic variables are assumed to be free, but it may happen that one or both are constrained to be nonnegative in the original model. This model was studied by Fukasawa and Günlük (2009), Dey and Wolsey (2008), and Conforti et al. (2009).

In this section, we consider the case where x_i is constrained to be nonnegative. We provide a closed-form formula to strengthen the cuts (7) for Type 1 and Type 2 triangles in canonical form. Contrary to Section 3.1, the results in this section do not apply to all Type 1 or 2 triangles, as some triangles can not be put into canonical form without affecting the constraint $x_i \geq 0$ in an important way. More precisely, this section applies to all triangles T of Type 1 or 2 with one side on the line $x_i = d$ for some integer d such that this side of T contains several integer points and $x_i \geq d+1$ (or $x_i \leq d-1$) is a constraint of problem (1). Indeed, considering the case $x_i \geq d+1$ (the other case is similar), the integer points in the interior of the other two sides of T are then $q^1 = (d+1, t)$ and $q^2 = (d+1, t+1)$ for some integer t. An integer translation maps q^1 and q^2 onto $(0,0)$ and $(0,1)$ respectively and maps the line $x_i = d+1$ onto $x_i = 0$. Then a unimodular transformation mapping line $x_i = a$ on itself for all integer a can be used to put the triangle in canonical form. Observe that the points satisfying $x_i \geq d+1$ are mapped by this transformation to the points satisfying $x_i \geq 0$. It follows that the results in this section apply to the triangles constructed in the algorithm presented in Section 2.2.

Intuitively, for a triangle as described in Section 2, if we have $x_i \geq 0$ then we can remove the vertical side of the triangle and expand it into an unbounded set containing no feasible solution of (1). This set is called a *wedge*.

We consider a Type 1 or 2 triangle in canonical form together with the feasible set, including the nonnegativity on x_i:

$$\begin{array}{rcl} x_i & = & f_i + \sum_{j=1}^{k} r_i^j s_j \\ x_\ell & = & f_\ell + \sum_{j=1}^{k} r_\ell^j s_j \\ x_i, x_\ell & \in & \mathbb{Z}, \\ x_i & \geq & 0 \\ s_j & \in & \mathbb{Z} \quad \text{for all} \quad j \in I \subseteq \{1, \ldots, k\}, \\ s_j & \geq & 0 \quad \text{for all} \quad j \in \{1, \ldots, k\}. \end{array} \tag{9}$$

The following disjunction is valid

$$(\eta x_i + x_\ell \geq 1) \quad \vee \quad (\mu x_i - x_\ell \geq 0) \ . \tag{10}$$

The intersection cut obtained from (10) is given by:

$$\sum_{j \in J} \max \left\{ \frac{\eta r_i^j + r_\ell^j}{1 - f_\ell - \eta f_i}, \frac{\mu r_i^j - r_\ell^j}{f_\ell - \mu f_i} \right\} s_j \geq 1. \tag{11}$$

Note that the intersection cut (11) dominates (7). In particular some of the coefficients of the latter cut may be negative. Balas (2009) observed the following.

Proposition 3.2. *Inequality* (11) *is valid for the split closure.*

Proof. Let P be the polyhedron obtained by dropping the integrality constraints in (9). Let $P_1 = P \cap \{x_\ell \geq 1\}$ and $P_2 = P \cap \{x_\ell \leq 0\}$. Inequality (11) is valid for $(P \cap \{\eta x_i + x_\ell \geq 1\}) \cup (P \cap \{\mu x_i - x_\ell \geq 0\})$. Since $P_1 \subseteq P \cap \{\eta x_i + x_\ell \geq 1\}$ and $P_2 \subseteq P \cap \{\mu x_i - x_\ell \geq 0\}$, inequality (11) is valid for $P_1 \cup P_2$, which proves that it is valid for the split closure of P. □

An optimal integer lifting for this cut can be computed in a similar way to the lifting computed in Section 3.2. The following proposition describes how the lifting can be adapted for disjunction (10). The difference with the lifting performed using Proposition B.1 is that here the disjunction can be shifted by any integer in the direction of the x_ℓ-axis but only by a positive integer in the direction of the x_i-axis. See Conforti et al. (2009) for a proof that this lifting is the unique minimal lifting.

Proposition 3.3. *For all $j \in I$, let $m^j \in \mathbb{Z}^2$ with $m_1^j \geq 0$. The disjunction*

$$\left(\eta\left(x_i + \sum_{j\in I} m_1^j s_j\right) + x_\ell - \sum_{j\in I} m_2^j s_j \geq 1\right) \quad \vee \quad \left(\mu\left(x_i + \sum_{j\in I} m_1^j s_j\right) - x_\ell + \sum_{j\in I} m_2^j s_j \geq 0\right) \tag{12}$$

is satisfied by all points $((x_i, x_\ell), s) \in \mathbb{Z}^2 \times \mathbb{R}^n$ with $x_i \geq 0$ and $s_j \in \mathbb{Z}_+ \ \forall j \in I$.

Proof. Let $((x_i, x_\ell), s) \in \mathbb{Z}^2 \times \mathbb{R}^n$ with $x_i \geq 0$ and $s_j \in \mathbb{Z}_+ \ \forall j \in I$. Suppose that $((x_i, x_\ell), s)$ does not satisfy (12). Let $x'_i = x_i + \sum_{j\in I} m_1^j s_j$ and $x'_\ell = x_\ell - \sum_{j\in I} m_2^j s_j$. Then $x'_i \in \mathbb{Z}_+$ and $x'_\ell \in \mathbb{Z}$ but (x'_i, x'_ℓ) does not satisfy (10), a contradiction. □

Using this proposition, we can derive a formula for a strengthened cut which uses the integrality of the variables s_j for $j \in I$. For $m_1 \in \mathbb{Z}_+$ and $m_2 \in \mathbb{Z}$, we define

$$\Psi^j(m_1, m_2) = \max\left\{\Phi_2^j(-m_1, m_2), \Phi_3^j(-m_1, m_2)\right\}.$$

The following cut is valid by application of Proposition 3.3:

$$\sum_{j\in I} \min_{(m_1,m_2)\in\mathbb{Z}_+\times\mathbb{Z}} \Psi^j(m_1, m_2)s_j + \sum_{j\in J\setminus I} \max\left\{\frac{\eta r_i^j + r_\ell^j}{1 - f_\ell - \eta f_i}, \frac{\mu r_i^j - r_\ell^j}{f_\ell - \mu f_i}\right\} s_j \geq 1 \tag{13}$$

The next proposition gives a closed form formula for computing the coefficients of this strengthened cut.

Proposition 3.4.

$$\min_{(m_1,m_2)\in\mathbb{Z}_+\times\mathbb{Z}} \Psi^j(m_1,m_2) = \min\left\{\frac{\eta r_i^j + r_\ell^j - \lfloor \overline{m}_2^j \rfloor}{1 - f_\ell - \eta f_i}, \frac{\mu r_i^j + \lceil \overline{m}_2^j \rceil - r_\ell^j}{f_\ell - \mu f_i}\right\}$$

with $\overline{m}_2^j = r_\ell^j + \frac{r_i^j(-\mu+(\mu+\eta)f_\ell)}{1-(\mu+\eta)f_i}$

Proof. Since η and μ are positive and $m_1 \geq 0$

$$\Phi_2^j(-m_1,m_2) = \frac{\eta\left(r_i^j + m_1\right) + r_\ell^j - m_2}{1 - \eta f_i - f_\ell} \geq \frac{\eta r_i^j + r_\ell^j - m_2}{1 - f_\ell - \eta f_i} = \Phi_2^j(0,m_2)$$
$$\Phi_3^j(-m_1,m_2) = \frac{\mu\left(r_i^j + m_1\right) - r_\ell^j + m_2}{f_\ell - \mu f_i} \geq \frac{\mu r_i^j - r_\ell^j + m_2}{f_\ell - \mu f_i} = \Phi_3^j(0,m_2)$$

and therefore

$$\min_{(m_1,m_2)\in\mathbb{Z}_+\times\mathbb{Z}} \Psi^j(m_1,m_2) = \min_{m_2\in\mathbb{Z}} \Psi^j(0,m_2).$$

Since $\Phi_2^j(0,m_2)$ is decreasing in m_2, $\Phi_3^j(0,m_2)$ is increasing in m_2 and $\Phi_2^j(0,\overline{m}_2^j) = \Phi_3^j(0,\overline{m}_2^j)$, the minimum is attained either in $\Phi_3^j(0,\lceil \overline{m}_2^j \rceil)$ or $\Phi_2^j(0,\lfloor \overline{m}_2^j \rfloor)$. □

Note that the cuts generated from the liftings of Propositions 3.1 and 3.4 are incomparable, i.e. neither is guaranteed to dominate the other.

4 Computational tests

We now present computational results obtained by applying the separation procedure devised in Section 2.2. The cut generator is implemented in C++ using the COIN-OR framework (Lougee-Heimer, 2003) with Cbc-2.2.2 and using Clp-1.8.2 for solving linear programs. The machine used for all the experiments in this section is a 64 bit `Monarch Empro 4-Way Tower Server` with four `AMD Opteron 852 2.6GHz` processors, each with eight `DDR-400` SDRAM of 2 `GB` and running `Linux Fedora 11`. The compiler is `gcc version 4.4.0 20090506 (Red Hat 4.4.0-4)`. Results are obtained using only one processor. Test instances for our experiments are the sixty-eight `MIPLIB3_C_V2` instances (Margot, 2009a). These instances are slight modifications of the standard `MIPLIB3` (Bixby et al., 1998) instances for which the validity of a provided feasible solution can be checked in finite precision arithmetic.

For many MILPs arising from practical applications, the LP relaxation has a degenerate optimal basis. For example, this is the case for 53 out of the 68 `MIPLIB3_C_V2` instances. Degenerate bases tend to be even more frequent after adding cuts. Traditional Gomory Mixed Integer (GMI) cuts from the tableau do not use the degenerate rows. Our main motivation is to determine whether using these degenerate rows as described in Section 2.2 produces cuts that are substantially different from GMI cuts, and if using them in conjunction with GMI cuts might be useful.

Dey et al. (2010) investigate the strength of Type 2 triangle cuts compared to GMI cuts from the tableau. On randomly generated multidimensional knapsack problems with up to five rows, they show that triangle cuts can substantially improve upon GMI cuts. However, they also mention that the number of triangle cuts they generate is huge. Since `MIPLIB3_C_V2` instances have typically much more than five rows, one needs to be very selective in the generation of two-row cuts to avoid extremely long computational times. As a consequence, our computational experiment gives a very different perspective on the empirical strength of two-row cuts.

4.1 Results on gap closed by our cuts

We compare the following four generators. As described in Section 2.2, we partition the rows of the optimal simplex tableau corresponding to the basic integer variables into fractional rows and integer rows.

- `G` : A cut generator which generates one GMI cut for each fractional row.
- `G-2rounds` : A cut generator which applies `G` to generate a first round of GMI cuts, reoptimizes the resulting LP and generates a second round of GMI cuts from the new optimal basic solution.
- `G+Allpairs` : A cut generator which generates one GMI cut for each fractional row and all cuts derived from every pair of fractional and integer rows as explained in Section 2.2. We also add cuts derived by removing the vertical side of the triangle whenever the variable x_i is at one of its bounds, as outlined in Section 3.3.
- `G+Deepest` : `G+Allpairs` often generates a large number of cuts. To reduce this number, we only add the GMI cuts and the deepest among all possible non-elementary split,

triangle and wedge cuts for each fractional row. By deepest cut, we mean the one with maximal Euclidean distance to the point to cut off in the space of the original variables. Note that `G` generates one cut per fractional row, whereas `G+Deepest` generates two.

Since computers work in finite precision, cut generators sometimes generate invalid cuts. To limit these occurrences, a variety of tolerances and safeguards are typically used. For GMI cuts, the most important ones are a lower bound on *integer infeasibility* and an upper bound on *dynamism*. For a solution $\bar{x}$, define the *integer infeasibility* of an integer variable x_i as $\min\{\bar{x}_i - \lfloor \bar{x}_i \rfloor, \lceil \bar{x}_i \rceil - \bar{x}_i\}$. We do not generate a cut from a fractional row whose basic integer variable has an integer infeasibility smaller than 0.01. Define the *dynamism* of a cut as the ratio between the largest and the smallest nonzero absolute values of its left hand side coefficients. We discard generated cuts whose dynamism is larger than 10^9. We use the same parameters for the generation of two-row cuts with, in addition, the condition that, for a row to be considered an integer row, the integer infeasibility of its basic integer variable must be at most 10^{-5}.

A method for testing the accuracy and strength of cut generators is given in Margot (2009b). Next we give a short description of the method. Please refer to the original paper for additional details.

Consider an instance I with a known feasible solution x^*. Define a *cutting step* as the operation of adding cuts obtained from a cut generator and checking if x^* is still feasible (report failure of the cut generator if x^* is no longer feasible). Define a *branching step* as getting an optimal solution $\bar{x}$ of the current LP relaxation, picking at random an integer variable x_i with a fractional value $\bar{x}_i$ and imposing in the LP that its value must be x^*_i.

The method, called a *dive towards a feasible solution* x^*, amounts to repeatedly performing a cutting step followed by a branching step until an integer feasible solution of the LP relaxation is obtained. In our experiments x^* is chosen to be a known feasible solution whose objective value is likely to be optimum or close to optimum. A number of dives (we use twenty dives in our experiments) are performed for each instance I. The solution x^* is the same in all dives. In our experiments, we set a time limit of 3 hours for each dive.

The motivation for using cuts in a branch-and-cut algorithm is to improve the bound obtained from the linear relaxation and to reduce the size of the enumeration tree. With these two goals in mind, we track the following during each dive:

- Fraction of the integrality gap closed after some given number of branching steps;
- Number of branching steps required to close some given fractions of the integrality gap.

The random branching step allows for meaningful statistical analysis of these two performance measures. We used the nonparametric *Quade* test as described in Conover (1999). All tests are done with a confidence level of 95%. The statistical package `R` (R Development Core Team, 2009) version 2.10.0 (2009-10-26) is used for the statistical analysis of the results.

In Table 1, we give the average gap closed by the four algorithms `G, G-2Rounds, G+Allpairs, G+Deepest` after 0, 4, 8 and 12 branching steps in a dive. The following problems timed out without any results for at least one generator: 10teams_c, air04_c, air05_c, arki_c, dano3mip_c, fast0507_c, mitre_c, mkc_c, seymour_c. The table gives results for the remaining 59 instances.

Table 1: Average percentage of gap closed at different depths

	Depth 0 (root node)	Depth 4	Depth 8	Depth 12
`G`	28.33	57.41	68.15	75.23
`G+Allpairs`	29.11	58.13	68.43	74.81
`G+Deepest`	28.80	58.09	68.59	75.40
`G-2Rounds`	36.66	59.41	68.75	75.47

Detailled tables are in the Appendix. They give, for each instance, the gap closed (Table 3) and the number of cuts generated at the root (Table 4), and the average (over the 20 dives) cpu time for completing a dive (Table 5).

We make the following observations from our results.

- Comparing `G` and `G+Allpairs`, we note that triangle cuts only improve marginally over the GMI cuts at depths 0, 4 and 8. The improvement in average gap closed at these three depths is less than 1%. At depth 12, the average gap closed is actually more for `G`, but this difference is not statistically significant according to the Quade test.

  Even restricting to instances that are degenerate at the root, the improvement is only 1.2% on average at the root which is disappointingly small given that we are generating at least four times as many cuts.

- Comparing `G+Allpairs` and `G+Deepest`, we note that most of the above improvement at depths 0, 4 and 8 can be achieved using only the deepest triangles. In fact, surprisingly, at depths 8 and 12 `G+Deepest` closes more average gap than `G+Allpairs`, and at depth 12 the difference is statistically significant. This may be due to the fact that adding too many cuts deteriorates the numerical properties of the basis, and this has an adverse effect on the quality of the cuts at greater depths.

- Comparing `G-2Rounds` and `G+Deepest`, which generate roughly the same number of cuts, we conclude that it is clearly better to use GMI cuts at levels 0 and 4. This is confirmed by the Quade test which shows that the difference is statistically significant. At depths 8 and 12 the difference is not statistically significant.

The results from Table 1 seem to indicate that our family of two-row cuts is not competitive with GMI cuts in terms of gap closed. This conclusion is confirmed by statistical tests performed on the second performance measure introduced above. For brevity, the detailed results of these tests are not reported here.

We performed other experiments which seem to suggest that among the two-row cuts that we generate, the split cuts are the more useful ones. First, we considered two variants of `G+Deepest` where the choice of the cut is based on the length of the segment p^2p^3 as defined in Section 2.2. In the first variant, we chose the cut with the longest segment p^2p^3; whereas, in the second we chose the shortest. As before, the cuts generated may be non-elementary split, triangle or wedge cuts. The motivation for the first choice comes from the theoretical results in Basu et al. (2009), where it is shown that the gap closed by the split closure is smaller when the segment p^2p^3 is long. However, surprisingly, the second variant closes more gap on average at levels 0, 4, 8 and 12, although the differences are small. This may be explained from the fact that the second variant generates more splits than the first and, in fact, at the root it generates almost only splits.

In another experiment, we tried to see whether our two-row cuts could cut off the optimum vertex of the split closure, as computed by Balas and Saxena (2008). For this purpose we used the formulations in `OSCLIB 1.0` (Saxena, 2008). In all but two of these instances, we could not cut off the split optimum.

We also compared the Euclidean depths of the GMI cuts and of our two-row cuts in the first round of cutting at the root node. For 15 instances, no triangle cuts are generated at the root because the optimal tableau is not degenerate and there are no basic integer variables with integer values. For the instance nw04_c, the average depths are too small for any meaningful comparison. For each of the 52 remaining instances and each fractional row ℓ, we measure the depth of the deepest triangle cut using row ℓ as a percentage of the depth of the GMI cut obtained from row ℓ. We then average this percentage over all fractional rows to get a single percentage number associated with that instance. For 46 out of the 52 instances, this percentage is between 99 and 125. The other 6 have a percentage between 125 and 200. This shows that, in general, the deepest triangle cuts are deeper than the corresponding GMI cuts.

4.2 Importance of the experimental setup

In this section, we show that one can reach drastically different conclusions by modifying the experimental setup in a seemingly natural way. Instead of using our own GMI cut generator, we now use the GMI cut generator `CglGomory` from the COIN-OR library with default settings. Table 2 displays the results corresponding to those in Table 1 when `CglGomory` is used instead of our GMI cut generator `G`.

Table 2: Average percentage of gap closed at different depths

	Depth 0 (root node)	Depth 4	Depth 8	Depth 12
`CglGomory`	23.83	51.86	62.91	69.83
`CglGomory+All pairs`	24.72	53.18	64.37	71.68
`CglGomory+Deepest`	24.45	53.92	64.54	71.19
`CglGomory-2Rounds`	30.89	53.73	62.85	69.97

The conclusions we reach from this table are different from those reached from Table 1. We conclude that `CglGomory+All pairs` is significantly better than `CglGomory` at all depths and this advantage increases with depth, with the difference being almost 2% at depth 12. These differences are statistically significant according to the Quade test. Moreover, `CglGomory+Deepest` is superior to `CglGomory-2Rounds` at depths 4, 8 and 12 with respect to gap closed and this difference is statistically significant at depths 8 and 12. When comparison is made on the number of branching steps needed to close different fractions of the gap, `CglGomory+Deepest` is significantly better than `CglGomory-2Rounds` for closing 70% or more of the gap according to the Quade test.

How do we reconcile the different conclusions reached from Tables 1 and 2 ?

Although our goal in this paper is to compare the strength of different cut families, in practice we can only compare cut generators. The generator `G` described in the previous section and `CglGomory` are two different generators for GMI cuts. We described above the parameter settings for accepting the cuts in `G`. For the default `CglGomory` generator the bound on integer infeasibility is 0.05, the bound on dynamism is 10^9, and there are several other parameters that affect the acceptance or rejection of a GMI cut. Even if we set the integer infeasibility and dynamism bounds to the same value in both generators, there are still significant differences in these two cut generators. This raises the question of which generator to use for our experiments and how to set the parameters. Depending on the parameter settings, a cut generator may reject cuts more often (conservative strategy) or less frequently (aggressive strategy).

A typical way that researchers test new cut generators is to add their generator to existing base generators and evaluate the difference in performance. In such a setting, if the base generator is conservative and the tested generator is aggressive, one may expect good results. On the other hand, if the base generator is made more aggressive or the tested generator is made more conservative, these improvements might be eroded to a large extent. The difference between Tables 1 and 2 is a concrete example of this phenomenon. This issue is usually not addressed in the literature. There is a trade-off between reliability of cut generators and their aggressiveness. Therefore, a key aspect of such experimental setups should involve measuring the reliability of cut generators. This was addressed by Margot (2009b). The general philosophy is to put the generator under stress and to record the number of failures due to invalid cut generation. Concretely, the reliability of a cut generator is tested using the diving experiment described in the previous section where 10 rounds of cuts are generated before a branching step is performed. According to this test, `G` is comparable in reliability to `CglGomory`. In fact, on the 59 instances of `MIPLIB3_C_V2` considered in this paper, `G` never cut the given feasible solution; whereas, `CglGomory` did cut it on harp2_c.

In our view, `G` should be used over `CglGomory` in our experiments because it is more aggressive and equally reliable. Another advantage of `G` over `CglGomory` is that not only are the parameter settings the same as those for the two-row cut generators, the computer code itself is the same. This does not necessarily imply that `G` should be preferred to `CglGomory` in a general branch-and-cut setting. More experiments would be needed to assess the relative empirical strength of the two cut generators.

This discussion shows that the setup based on `CglGomory` is flawed and that the correct setup is the one used in Section 4.1, if the goal is to evaluate our two-row cuts as compared to GMI cuts. The conclusion is that our family of two-row cuts is not competitive with GMI cuts.

Acknowledgements

This research was supported by a Mellon Fellowship, ANR grant ANR06-BLAN-0375, NSF grant CMMI0653419 and ONR grant N00014-03-1-0133.

References

Andersen, K., Q. Louveaux, R. Weismantel, L. Wolsey. 2007. Cutting planes from two rows of a simplex tableau. *Proceedings of IPCO XII*, *Lecture Notes in Computer Science*, vol. 4513. 1–15.

Balas, E. 1971. Intersection cuts - a new type of cutting planes for integer programming. *Operations Research* **19** 19–39.

Balas, E. 2009. Personal communication.

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

Balas, E., A. Saxena. 2008. Optimizing over the split closure. *Mathematical Programming A* **113** 219–240.

Basu, A., P. Bonami, G. P. Cornuéjols, F. Margot. 2009. On the relative strength of split, triangle and quadrilateral cuts. *To Appear in Mathematical Programming* DOI:10.1007/s10107-009-0281-x.

Bixby, R.E., S. Ceria, C.M. McZeal, M.W.P Savelsbergh. 1998. MIPLIB 3.0. `http://www.caam.rice.edu/∼bixby/miplib/miplib.html`.

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

Conforti, M., G. Cornuéjols, G. Zambelli. 2009. A geometric perspective on lifting. Technical report.

Conover, W.J. 1999. *Practical Nonparametric Statistics, 3rd edition*. Wiley.

Cook, W., R. Kannan, A. Schrijver. 1990. Chvátal closures for mixed integer programming problems. *Mathematical Programming* **47** 155–174.

Cornuéjols, G., F. Margot. 2009. On the facets of mixed integer programs with two integer variables and two constraints. *Mathematical Programming* **120** 419–456.

Crowder, H., E.L. Johnson, M. Padberg. 1983. Solving large-scale zero-one linear programming problems. *Operations Research* **31** 803–834.

Dey, S. S., A. Lodi, A. Tramontani, L. A. Wolsey. 2010. Experiments with two row tableau cuts. *Proceedings of IPCO 2010*, *Lecture Notes in Computer Science*, vol. 6080. 424–437.

Dey, S.S., L.A. Wolsey. 2008. Lifting integer variables in minimal inequalities corresponding to lattice-free triangles. *Proceedings of IPCO 2008*, *Lecture Notes in Computer Science*, vol. 5035. 463–476.

Dey, S.S., L.A. Wolsey. 2010. Two row mixed-integer cuts via lifting. *To Appear in Mathematical Programming* DOI: 10.1007/s10107-010-0362-x.

Espinoza, D.G. 2008. Computing with multirow gomory cuts. *Proceedings of IPCO 2008*, *Lecture Notes in Computer Science*, vol. 5035. 214–224.

Fukasawa, R., O. Günlük. 2009. Strengthening lattice-free cuts using non-negativity. Available on OptimizationOnline.

Gomory, R.E. 1963. An Algorithm for Integer Solutions to Linear Programs, *Recent Advances in Mathematical Programming*, R.L. Graves and P. Wolfe eds., McGraw-Hill, New York, 269–302.

Gomory, R.E. 1969. Some polyhedra related to combinatorial problems. *Linear Algebra and its Applications* **2** 451–558.

Gomory, R.E., E.L. Johnson. 1972. Some continuous functions related to corner polyhedra, part I. *Mathematical Programming* **3** 23–85.

Johnson, E.L. 1974. On the group problem for mixed integer programming. *Mathematical Programming Study* **2** 137–179.

Lougee-Heimer, R. 2003. The common optimization interface for operations research. *IBM Journal of Research and Development* **47** 57–66. `http://www.coin-or.org`.

Lovász, L. 1989. Geometry of Numbers and Integer Programming, *Mathematical Programming: Recent Developments and Applications*, M. Iri and K. Tanabe eds., Kluwer, 177–210.

Marchand, H., L.A. Wolsey. 2001. Aggregation and mixed integer rounding to solve MIPs. *Operations Research* **49** 363–371.

Margot, F. 2009a. MIPLIB3_C_V2. `http://wpweb2.tepper.cmu.edu/fmargot/`.

Margot, F. 2009b. Testing cut generators for mixed integer linear programming. *Mathematical Programming Computation* **1** 69–95.

Nemhauser, G.L., L.A. Wolsey. 1990. A recursive procedure to generate all cuts for 0-1 mixed integer programs. *Mathematical Programming* **46** 379–390.

R Development Core Team. 2009. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL `http://www.R-project.org`. ISBN 3-900051-07-0.

Saxena, A. 2008. OSCLIB. `http://www.andrew.cmu.edu/user/anureets/osc/osc.htm.`

A Intersection cuts from multiple rows

Proposition A.1 gives the general formula for the cut derived from a p-term disjunction Π and m rows of a simplex tableau.

Proposition A.1. *Let* $C = \{(x, s) \in \mathbb{Z}^m \times \mathbb{R}^k : x = f + \sum_{j=1}^{k} r^j s_j, s \geq 0\}$ *and*

$$\Pi = \left(\pi^1 x \geq \pi_0^1\right) \vee \ldots \vee \left(\pi^p x \geq \pi_0^p\right)$$

be a disjunction satisfied by all points of $\mathbb{Z}^m$ *and not satisfied by* f*. The inequality*

$$\sum_{j=1}^{k} \max_{l=1,\ldots,p} \left\{ \frac{\pi^l r^j}{\pi_0^l - \pi^l f} \right\} s_j \geq 1 \tag{14}$$

is valid for C*.*

Proof. For each $l = 1, \ldots, p$, $\pi^l x \geq \pi_0^l$ can be rewritten using the definition of C in terms of the s variables only:

$$\pi^l \left(f + \sum_{j=1}^{k} r^j s_j \right) \geq \pi_0^l. \qquad\qquad l = 1 \ldots, p.$$

Reorganizing the terms of the above inequality it can be rewritten as

$$\sum_{j=1}^{k} \pi^l r^j s_j \geq \pi_0^l - \pi^l f. \qquad\qquad l = 1 \ldots, p.$$

The point f does not satisfy the disjunction and therefore $\pi_0^l - \pi^l f > 0$. Dividing the inequality by $\pi_0^l - \pi^l f$ we obtain

$$\sum_{j=1}^{k} \frac{\pi^l r^j}{\pi_0^l - \pi^l f} s_j \geq 1 \qquad\qquad l = 1 \ldots, p.$$

Each of the above inequalities is valid for one term of the disjunction. Since they all have the same right-hand-side and $s \geq 0$, the inequality obtained by taking the component-wise maximum is valid for all terms of the disjunction and therefore

$$\sum_{j=1}^{k} \max_{l=1,\ldots,p} \left\{ \frac{\pi^l r^j}{\pi_0^l - \pi^l f} \right\} s_j \geq 1$$

is valid for C. □

The inequality is nothing more than the intersection cut obtained from the cone C and the disjunction Π.

B Integer lifting by the trivial fill-in function

Proposition B.1. *Let $C = \{(x,s) \in \mathbb{Z}^m \times \mathbb{R}^k : x = f + \sum_{j=1}^k r^j s_j, s \geq 0\}$. Let I be a subset of $\{1,\ldots,k\}$, $C^I = C \cap \left\{(x,s) \in \mathbb{Z}^m \times \mathbb{R}^k : s_j \in \mathbb{Z}, \forall j \in I\right\}$ be a mixed-integer set and*

$$\Pi = \left(\pi^1 x \geq \pi_0^1\right) \vee \ldots \vee \left(\pi^p x \geq \pi_0^p\right)$$

be a disjunction satisfied by all $x \in \mathbb{Z}^m$. For $j \in I$, let m^j be an integral vector of dimension m. The disjunction

$$\Pi' = \left(\pi^1\left(x - \sum_{j\in I} m^j s_j\right) \geq \pi_0^1\right) \vee \ldots \vee \left(\pi^p\left(x - \sum_{j\in I} m^j s_j\right) \geq \pi_0^p\right)$$

is valid for C^I.

Proof. Suppose that the disjunction Π' is not valid. Then there exist $(x,s) \in \mathbb{Z}^m \times \mathbb{R}^k$ with $s_j \in \mathbb{Z}$, $\forall j \in I$, such that:

$$\pi^l(x - \sum_{j\in I} m^j s_j) < \pi_0^l \qquad\qquad l = 1,\ldots,p.$$

Let $x' = x - \sum_{j\in I} m^j s_j$. $x' \in \mathbb{Z}^m$ since $x \in \mathbb{Z}^m$, $m^j \in \mathbb{Z}^m$ and $s_j \in \mathbb{Z}$ for all $j \in I$. Then x' violates disjunction Π which contradicts the fact that Π is a valid disjunction. $\square$

Now we consider the intersection cut obtained from a disjunction of the form Π' stated in Proposition B.1.

Proposition B.2. *Let $C = \{(x,s) \in \mathbb{Z}^m \times \mathbb{R}^k : x = f + \sum_{j=1}^k r^j s_j, s \geq 0\}$. Let $J = \{1,\ldots,k\}$ and I be a subset of J, $C^I = C \cap \left\{(x,s) \in \mathbb{Z}^m \times \mathbb{R}^k : s_j \in \mathbb{Z}, \forall j \in I\right\}$ be a mixed-integer set and*

$$\Pi = (\pi^1 x \geq \pi_0^1) \vee \ldots \vee (\pi^p x \geq \pi_0^p)$$

be a disjunction satisfied by all $x \in \mathbb{Z}^m$ and not satisfied by the point f. For all $m^j \in \mathbb{Z}^m$, $j = 1,\ldots,m$ the inequality

$$\sum_{j\in I} \max_{l=1,\ldots,p} \left\{\frac{\pi^l(r^j - m^j)}{\pi_0^l - \pi^l f}\right\} s_j + \sum_{j\in J\setminus I} \max_{l=1,\ldots,p} \left\{\frac{\pi^l r^j}{\pi_0^l - \pi^l f}\right\} s_j \geq 1 \tag{15}$$

is valid for C^I.

Proof. Since Π is valid, for all $m^j \in \mathbb{Z}^m$, $j = 1, \ldots, m$ the disjunction

$$\Pi' = (\pi^1(x - \sum_{j \in I} m^j s_j) \geq \pi_0^1) \vee \ldots \vee (\pi^k(x - \sum_{j \in I} m^j s_j) \geq \pi_0^p)$$

is valid.

The terms of the disjunction can be rewritten as:

$$\sum_{j \in I} \pi^l (r^j - m^j) s_j + \sum_{j \in J \setminus I} \pi^l r^j s_j \geq \pi_0^l - \pi^l f. \qquad l = 1 \ldots, p.$$

The point f does not satisfy the disjunction and therefore $\pi_0^l - \pi^l f > 0$. Dividing the inequality by $\pi_0^l - \pi^l f$ we obtain

$$\sum_{j \in I} \frac{\pi^l (r^j - m^j)}{\pi_0^l - \pi^l f} s_j + \sum_{j \in J \setminus I} \frac{\pi^l r^j}{\pi_0^l - \pi^l f} s_j \geq 1 \qquad l = 1 \ldots, p.$$

Each of the above inequalities is valid for one term of the disjunction. By applying the disjunctive principle we obtain (15). □

C Detailed tables of computational results

Prob	G	G+Allpairs	G+Deepest	G-2Rounds
air03_c	99.92	99.92	99.92	99.92
bell3a_c	44.88	44.88	44.88	66.55
bell4_c	23.37	23.37	23.37	45.10
bell5_c	14.53	14.70	14.70	17.93
blend2_c	15.99	16.06	15.99	17.43
cap6000_c	39.91	39.91	39.91	55.45
danoint_c	0.26	0.26	0.26	0.47
dcmulti_c	39.62	40.21	39.98	51.98
dsbmip_mod_c	0.00	0.00	0.00	0.00
egout_c	35.95	35.95	35.95	61.66
enigma_c	100.00	100.00	100.00	100.00
fiber_c	72.63	72.63	72.63	80.38
fixnet3_c	49.38	49.38	49.38	64.27
fixnet4_c	22.54	22.54	22.54	39.36
fixnet6_c	10.74	10.74	10.74	26.08
flugpl_c	11.74	11.74	11.74	13.21
gen_c	60.38	60.38	60.38	64.22
gesa2_c	28.04	28.74	28.12	59.89
gesa2_o_c	30.79	33.55	33.28	59.77
gesa3_c	47.49	47.49	47.49	50.26
gesa3_o_c	50.49	50.49	50.49	53.58
gt2_c	91.87	91.87	91.87	95.98
harp2_c	23.81	27.01	23.81	30.16
khb05250_c	74.90	74.90	74.90	88.34

Prob	G	G+Allpairs	G+Deepest	G-2Rounds
l152lav_c	5.75	13.93	13.86	11.10
lseu_c	55.19	55.83	55.83	60.73
markshare1_c	0.00	0.00	0.00	0.00
markshare2_c	0.00	0.00	0.00	0.00
mas74_c	6.69	6.69	6.69	7.45
mas76_c	6.43	6.43	6.43	6.78
misc03_c	6.12	7.24	6.12	6.38
misc06_c	0.42	0.42	0.42	3.63
misc07_c	0.72	0.72	0.72	0.72
mod008_c	20.10	20.86	20.86	22.36
mod010_c	99.94	99.94	99.94	99.94
mod011_c	29.19	32.89	29.92	35.74
modglob_c	16.96	16.96	16.96	40.50
noswot_c	0.00	0.00	0.00	0.00
nw04_c	56.82	66.07	56.82	97.08
p0033_c	56.82	57.03	56.82	57.02
p0201_c	33.78	33.78	33.78	44.66
p0282_c	3.70	3.70	3.70	8.69
p0548_c	40.45	40.45	40.45	57.84
p2756_c	0.21	0.21	0.21	0.29
pk1_c	0.00	0.00	0.00	0.00
pp08a_c	54.69	54.69	54.69	68.41
pp08aCUTS_c	33.79	33.79	33.79	45.42
qiu_c	2.89	2.89	2.89	10.75
qnet1_c	12.79	14.05	12.79	20.02
qnet1_o_c	39.74	42.08	41.83	47.59
rentacar_c	4.07	4.07	4.07	8.37
rgn_c	6.39	6.39	6.39	11.48
rout_c	0.32	0.63	0.32	2.38
set1ch_c	39.09	39.09	39.09	75.02
stein27_c	0.00	0.00	0.00	0.00
stein45_c	0.00	0.00	0.00	0.00
swath_c	8.01	8.28	8.18	14.14
vpm1_c	26.18	37.09	37.09	35.45
vpm2_c	15.08	16.62	16.62	21.05

Table 3: Gap closed at the root.

Prob	G	G+Allpairs	G+Deepest	G-2Rounds
air03_c	36	6291	72	36
bell3a_c	23	25	24	29
bell4_c	45	46	46	76
bell5_c	24	25	25	40
blend2_c	6	190	12	17
cap6000_c	2	425	4	5
danoint_c	31	31	31	61
dcmulti_c	49	165	65	96
dsbmip_mod_c	56	528	92	92

Prob	G	G+Allpairs	G+Deepest	G-2Rounds
egout_c	27	27	27	49
enigma_c	6	185	12	10
fiber_c	43	1936	84	75
fixnet3_c	41	41	41	69
fixnet4_c	38	38	38	82
fixnet6_c	33	33	33	65
flugpl_c	10	10	10	18
gen_c	42	91	64	70
gesa2_c	57	436	78	102
gesa2_o_c	71	726	104	130
gesa3_c	83	3071	137	137
gesa3_o_c	96	3518	157	159
gt2_c	11	11	11	26
harp2_c	30	2788	60	81
khb05250_c	19	20	20	33
l152lav_c	53	6327	106	118
lseu_c	12	35	19	22
markshare1_c	6	6	6	12
markshare2_c	6	6	6	13
mas74_c	12	12	12	29
mas76_c	11	11	11	25
misc03_c	12	549	24	39
misc06_c	2	2	2	4
misc07_c	26	1519	52	60
mod008_c	5	18	10	11
mod010_c	35	10018	70	35
mod011_c	16	154	32	47
modglob_c	25	45	34	55
noswot_c	21	27	26	35
nw04_c	6	508	12	14
p0033_c	6	25	11	14
p0201_c	20	331	38	62
p0282_c	26	37	32	47
p0548_c	39	50	40	86
p2756_c	10	10	10	24
pk1_c	15	15	15	30
pp08a_c	53	88	74	101
pp08aCUTS_c	46	104	62	91
qiu_c	36	696	68	81
qnet1_c	50	15014	100	103
qnet1_o_c	11	615	22	43
rentacar_c	12	12	12	19
rgn_c	19	19	19	36
rout_c	34	499	67	63
set1ch_c	136	268	148	253
stein27_c	21	21	21	43
stein45_c	35	35	35	70
swath_c	45	6321	90	93
vpm1_c	21	111	32	33
vpm2_c	31	71	56	68

Table 4: Number of cuts at the root.

Prob	G	G+Allpairs	G+Deepest	G-2Rounds
air03_c	0.19	76.48	41.02	0.25
bell3a_c	0.01	0.08	0.04	0.02
bell4_c	0.05	0.28	0.15	0.09
bell5_c	0.02	0.08	0.05	0.04
blend2_c	0.12	3.34	1.18	0.23
cap6000_c	38.23	279.64	147.17	29.77
danoint_c	5.50	62.07	8.80	9.33
dcmulti_c	0.29	6.09	2.21	0.66
dsbmip_mod_c	5.20	145.80	27.89	9.29
egout_c	0.02	0.12	0.06	0.05
enigma_c	0.01	0.13	0.08	0.02
fiber_c	0.34	324.85	47.10	0.76
fixnet3_c	0.10	2.03	0.39	0.12
fixnet4_c	1.85	97.19	25.47	3.63
fixnet6_c	1.55	73.51	20.89	2.92
flugpl_c	0.00	0.00	0.00	0.00
gen_c	0.09	2.79	0.82	0.18
gesa2_c	0.35	6.20	2.24	0.62
gesa2_o_c	0.46	24.04	6.27	0.87
gesa3_c	0.14	3.71	1.52	0.31
gesa3_o_c	0.17	9.14	2.97	0.37
gt2_c	0.04	0.21	0.19	0.07
harp2_c	10.17	900.40	573.15	19.86
khb05250_c	0.07	0.60	0.25	0.14
l152lav_c	1.89	147.80	75.46	5.09
lseu_c	0.01	0.07	0.05	0.02
markshare1_c	0.02	0.03	0.03	0.04
markshare2_c	0.04	0.05	0.04	0.06
mas74_c	0.08	0.12	0.09	0.14
mas76_c	0.06	0.09	0.06	0.10
misc03_c	0.05	2.49	0.66	0.11
misc06_c	0.04	0.07	0.06	0.04
misc07_c	0.21	14.12	2.99	0.39
mod008_c	0.02	0.03	0.03	0.04
mod010_c	0.10	33.68	13.79	0.12
mod011_c	54.95	697.54	213.95	97.26
modglob_c	0.19	6.91	1.90	0.36
noswot_c	0.10	1.45	0.54	0.18
nw04_c	4.74	334.90	119.57	4.93
p0033_c	0.00	0.01	0.01	0.00
p0201_c	0.06	7.92	1.07	0.15
p0282_c	0.13	0.62	0.45	0.27
p0548_c	0.95	75.07	25.63	1.64
p2756_c	1.60	5.38	3.26	3.17
pk1_c	0.07	0.09	0.08	0.12
pp08a_c	0.33	2.64	1.15	0.55
pp08aCUTS_c	0.39	3.47	1.48	0.78
qiu_c	2.75	12.85	4.62	5.29
qnet1_c	6.34	2607.14	483.16	10.94
qnet1_o_c	3.71	2147.31	509.33	10.53
rentacar_c	6.78	8.97	8.95	10.28
rgn_c	0.02	0.06	0.05	0.04

Prob	`G`	`G+Allpairs`	`G+Deepest`	`G-2Rounds`
rout_c	0.52	11.26	4.59	0.93
set1ch_c	1.73	54.50	14.89	3.33
stein27_c	0.02	0.04	0.03	0.03
stein45_c	0.11	0.23	0.17	0.22
swath_c	6.36	1846.57	1101.84	15.07
vpm1_c	0.09	0.47	0.23	0.14
vpm2_c	0.47	2.53	1.93	1.02

Table 5: Average cpu time per dive.