IBM Research Report

Computational Experiments with Cross and Crooked Cross Cuts

Sanjeeb Dash, Oktay Günlük
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Juan Pablo Vielma
University of Pittsburgh

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Computational Experiments with Cross and Crooked Cross Cuts

Sanjeeb Dash
IBM Research
sanjeebd@us.ibm.com

Oktay Günlük
IBM Research
gunluk@us.ibm.com

Juan Pablo Vielma
University of Pittsburgh
jvielma@pitt.edu

June 22, 2011

Abstract

In a recent paper, Dash, Dey and Günlük (2010) showed that many families of inequalities for the two-row continuous group relaxation and variants of this relaxation are cross cuts or crooked cross cuts, both of which generalize split cuts. Li and Richard (2008) recently studied t-branch split cuts for mixed-integer programs for integers $t \geq 1$. Split cuts are just 1-branch split cuts, and cross cuts are 2-branch split cuts. In this paper, we study whether cross and crooked cross cuts can be separated in an effective manner for practical MIPs, and can yield a non-trivial improvement over the bounds obtained by split cuts. We also study whether such cuts obtained from two simplex tableau rows at a time can strengthen the bounds obtained by GMI cuts based on single tableau rows. We give positive answers to both these questions for MIPLIB 3.0 problems.

1 Introduction

There has been much recent work on the use of lattice-free sets to generate cutting planes (*lattice-free cuts*) from multiple rows of a simplex tableau associated with a mixed-integer program (MIP). Andersen, Louveaux, Weismantel and Wolsey [1] studied the *two-row continuous group relaxation*, a set of the form

$$W = \{(z, s) \in \mathbb{Z}^2 \times \mathbb{R}^n_+ : z_1 + r_1 s = f_1, z_2 + r_2 s = f_2\}, \tag{1}$$

where $r_1, r_2 \in \mathbb{R}^{1 \times n}$ and $f_1, f_2 \in \mathbb{R}$, and showed that the convex hull of its solutions is given by cuts obtained from polyhedral lattice-free sets in $\mathbb{R}^2$ (i.e., *2D lattice-free* cuts) with at most 4 sides. Subsequently, others have studied extensions to the semi-infinite version of the k-row continuous group relaxation for $k \geq 2$ [14], and extensions with more structure such as the integrality of non-basic variables [27], the nonnegativity of basic integer variables [10], [27], [31], and both the integrality of non-basic variables and nonnegativity of basic integer variables [9] [15]. See [25] and [16] for recent surveys on this topic.

Balas [3] proposed comparing lattice-free cuts for two-row continuous group relaxations to t-branch split cuts defined by Li and Richard [33]). Dash, Dey, and Günlük [18] [19] introduced *crooked cross cuts* for general mixed integer sets, and showed that they dominate 2-branch split cuts (which they call *cross cuts*). They further showed that all valid inequalities (i.e., 2D lattice free cuts) for the two-row continuous group relaxation are crooked cross cuts, and that many other cuts in the literature for variants of the two-row continuous group relaxation are also crooked cross cuts. In particular, cuts derived in [27], [10], [31], [9], and [15] for variants of the set W where the variables z_1 and z_2 have upper and lower bounds, or where some of the s variables also have integrality restrictions are crooked cross cuts. However, the problem of finding violated cross and crooked cuts effectively for practical MIPs is not addressed in [18] [19], and we

1

study this computational problem here. Further, cross and crooked cross cuts can be viewed as generalizing both cuts obtained from two-row continuous group relaxations and split cuts, and we explore both these aspects.

The Gomory mixed-integer (GMI) cut can be seen as a lattice-free cut derived from a *one-row continuous group relaxation* (a set defined by using only one of the two equality constraints of W) which is then strengthened using the integrality information on the s variables. GMI cuts generated from rows of the optimal simplex tableau associated with the LP relaxation of an MIP are known to be very effective in practice and are now incorporated in most commercial MIP solvers. An important motivation for the recent interest in the two-row continuous group relaxation is the hope that cuts obtained from pairs of tableau rows could significantly improve the bounds yielded by GMI cuts. Taking two simplex tableau rows corresponding to basic integer variables, and then relaxing the nonnegativity of the basic variables as well as the integrality of the non-basic variables yields such a relaxation. In the first work on this topic, Espinoza [28] generated some (lattice-free) cuts from k tableau rows, for k up to 10. Further, he generated multiple rounds of such cuts and compared their effect with multiple rounds of GMI cuts. His work does not measure the relative strength of one round of GMI cuts versus one round of all possible cuts from multiple tableau rows. Louveaux and Poirrier [34] give a fast algorithm to separate all lattice-free cuts from two-row relaxations arising from pairs of tableau rows. Like Espinoza, they consider a subset of all possible tableau row pairs and generate lattice-free cuts, augment the LP relaxation with these cuts and derive a new tableau (up to five times), but they do not lift the cuts with respect to the non-basic integral variables. Recently, Dey, Lodi, Tramontani and Wolsey [24] experimented with cuts obtained from pairs of tableau rows. They

(i) construct the two-row continuous group relaxation from a given pair of tableau rows by relaxing the integrality of non-basic integral variables,

(ii) find violated 2D lattice-free cuts, in particular triangle cuts of type 2, and

(iii) then lift the coefficients of the variables which were relaxed to be continuous, so as to re-introduce integrality information.

Their computational experiments are performed on randomly generated problems, where they get a nontrivial improvement over one round of GMI cuts, but their procedure has limited success on practical MIPs. Basu, Bonami, Cornuéjols, Margot [8] also separate triangle cuts of type 2 (motivated in part by their results in [7]), but conclude that their "family of two-row cuts is not competitive with GMI cuts in terms of gap closed" for MIPLIB problems. Their approach also consists of steps (i)–(iii) above, and this approach has some inherent difficulties. Firstly, it is not always clear what the best way to lift a given lattice-free cut is. Secondly, it is hard to decide which lattice-free cut for the two-row continuous group relaxation would yield a good cut after lifting. Therefore, although Louveaux and Poirrier [34] have shown that lattice free cuts from such a relaxation can be separated quickly, it is still not clear if this relaxation is the best model for obtaining cuts from pairs of tableau rows.

As discussed earlier, given a pair of tableau rows, the cuts obtained by steps (i)–(iii) above are crooked cross cuts obtained from the tableau rows [18]. Further, if a nonbasic variable has finite upper and lower bounds, then it is not known if all crooked cross cuts are obtainable using lifted lattice-free cuts. These facts suggest an alternative to the approaches in [8] and [24] to finding cuts for tableau row pairs, namely finding violated cross cuts and crooked cross cuts. We devise heuristics to find such cuts, and apply them to pairs of rows from the optimal simplex tableau of the LP relaxation of an MIP. We demonstrate that one can obtain bounds which are significantly stronger than the bounds obtained by only adding GMI cuts based on these tableau rows.

Recently, Balas and Saxena [6] and Dash, Günlük and Lodi [23] approximately optimized over the split closure of practical MIP instances and obtained very strong bounds on the optimal values of such instances. As cross and crooked cross cuts generalize split cuts, we are also interested in comparing the bounds obtained by the former class of cuts with the bounds in [6],[23], obtained using split cuts. For a number of the MIPs in the MIPLIB 3.0 library, we are able to obtain noticeably better bounds on the integer optimum value by generating cross cuts.

The remainder of the paper is structured as follows. In Section 2, we define the different classes of cuts used in this paper, and in Section 3, we discuss separation models for these cut classes. In Section 4, we describe the main heuristics we use to separate cuts from these classes. In Section 5, we use the heuristics described in the previous sections to obtain cross cuts from pairs of tableau rows and show that these cuts yield significantly better bounds than the GMI cuts derived from the same tableau rows. In Section 6, we explain how we find violated cross cuts and crooked cross cuts (from all constraints). We compare bounds on the integer optimum value obtained in this manner with bounds obtained earlier by approximately optimizing over the split closure.

2 Preliminaries

Consider the following mixed-integer set with m rows

$$P = \{(x,y) \in \mathbb{Z}^{n_1} \times \mathbb{R}^{n_2} \ : \ Ax + Gy = b, \ x \geq 0, \ y \geq 0\}$$

where A, G, b are rational matrices with m rows and $n_1, n_2, 1$ columns, respectively. In general, the set of feasible solutions for any mixed-integer linear program can be framed in this way. We denote the linear programming (LP) relaxation of P by P^{LP}. In the remainder of the paper π and a (along with subscripts or superscripts) represent row vectors with n_1 components, and c is a row vector with n_2 components. We next discuss disjunctive cuts for P (see Balas [2]).

Let $D = \cup_{k \in K} D_k$ where $D_k = \{(x,y) \in \mathbb{Z}^{n_1} \times \mathbb{R}^{n_2} : A^k x + G^k y \leq b^k\}$ for $k \in K$. If $\mathbb{Z}^{n_1} \times \mathbb{R}^{n_2} \subseteq D$, then we call D a *disjunction* and we call each D_k an *atom* of the disjunction D. A linear inequality is called a *disjunctive cut* for P obtained from the disjunction D if it is valid for $P^{LP} \cap D_k$ for all $k \in K$. All disjunctive cuts for P are valid for P. Note that multiple disjunctive cuts can be derived from the same disjunction. In this paper we are interested in the following types of disjunctions.

2.1 Split disjunctions

For a fixed $\pi \in \mathbb{Z}^{n_1} \setminus \{\mathbf{0}\}$, and $\gamma \in \mathbb{Z}$, a *split disjunction* is denoted by $S(\pi, \gamma)$ and defined as

$$S(\pi, \gamma) = S_1(\pi, \gamma) \cup S_2(\pi, \gamma),$$

where $S_1(\pi, \gamma) = \{(x,y) \in \mathbb{R}^{n_1+n_2} \ : \ \pi x \leq \gamma\}$ and $S_2(\pi, \gamma) = \{(x,y) \in \mathbb{R}^{n_1+n_2} \ : \ \pi x \geq \gamma + 1\}$. A linear inequality is said to be a *split cut* [17] for P if it is valid for $P^{LP} \cap S_1(\pi, \gamma)$ and $P^{LP} \cap S_2(\pi, \gamma)$. We define $P_S(\pi, \gamma)$ as the convex hull of $P^{LP} \cap S(\pi, \gamma)$, and P_S to be the set of points in P^{LP} which satisfy all split cuts for P, i.e., the *split closure* of P. Therefore

$$P_S = \bigcap_{\pi \in \mathbb{Z}^{n_1}} \bigcap_{\gamma \in \mathbb{Z}} P_S(\pi, \gamma). \tag{2}$$

Assume that the following equation is satisfied by all points in P (for example, it could be obtained via a linear combination of the constraints $Ax + Gy = b$):

$$\sum_{i=1}^{n_1} a_i x_i + \sum_{i=1}^{n_2} c_i y_i = \beta. \tag{3}$$

Let $\hat{\beta} = \beta - \lfloor \beta \rfloor$ and assume that $\hat{\beta} \neq 0$. The *mixed-integer rounding* (MIR) cut derived from (3) is

$$\sum_{i=1}^{n_1} (\hat{\beta} \lfloor a_i \rfloor + \min(\hat{a}_i, \hat{\beta})) x_i + \sum_{i=1}^{n_2} \max(c_i, 0) x_i \geq \hat{\beta} \lceil \beta \rceil, \tag{4}$$

where $\hat{a}_i = a_i - \lfloor a_i \rfloor$. It is well known that the MIR cut (4) is a split cut for the following 1-row relaxation of P,

$$P_1 = \{(x, y) \in \mathbb{Z}^{n_1} \times \mathbb{R}^{n_2} \; : \; \sum_{i=1}^{n_1} a_i x_i + \sum_{i=1}^{n_2} c_i y_i = \beta, \; x \geq 0, \; y \geq 0\},$$

using the disjunction $S(\pi, \gamma)$ where

$$\gamma = \lfloor \beta \rfloor \text{ and } \pi_i = \begin{cases} \lfloor a_i \rfloor & \text{if } \hat{a}_i \leq \hat{\beta}, \\ \lceil a_i \rceil & \text{if } \hat{a}_i > \hat{\beta}. \end{cases} \tag{5}$$

Conversely, any split cut for P is an MIR cut derived from a single implied equation (3) for P along with nonnegativity constraints on the variables. When the MIR cut is derived from a row of the simplex tableau associated with the LP relaxation of the mixed-integer program, we call it the *Gomory mixed-integer* (GMI) cut.

2.2 Cross disjunctions and cross cuts

The *cross disjunction* (see [18]) associated to $\pi_1, \pi_2 \in \mathbb{Z}^{n_1} \setminus \{\mathbf{0}\}$, and $\gamma_1, \gamma_2 \in \mathbb{Z}$, is given by

$$C(\pi_1, \pi_2, \gamma_1, \gamma_2) = \bigcup_{k \in \{1,2,3,4\}} C_k(\pi_1, \pi_2, \gamma_1, \gamma_2)$$

where:

$$C_1(\pi_1, \pi_2, \gamma_1, \gamma_2) = \{(x, y) \in \mathbb{R}^{n_1+n_2} \; : \; \pi_1 x \leq \gamma_1, \; \pi_2 x \leq \gamma_2\},$$
$$C_2(\pi_1, \pi_2, \gamma_1, \gamma_2) = \{(x, y) \in \mathbb{R}^{n_1+n_2} \; : \; \pi_1 x \leq \gamma_1, \; \pi_2 x \geq \gamma_2 + 1\},$$
$$C_3(\pi_1, \pi_2, \gamma_1, \gamma_2) = \{(x, y) \in \mathbb{R}^{n_1+n_2} \; : \; \pi_1 x \geq \gamma_1 + 1, \; \pi_2 x \leq \gamma_2\}, \text{ and}$$
$$C_4(\pi_1, \pi_2, \gamma_1, \gamma_2) = \{(x, y) \in \mathbb{R}^{n_1+n_2} \; : \; \pi_1 x \geq \gamma_1 + 1, \; \pi_2 x \geq \gamma_2 + 1\}.$$

Cross disjunctions were first defined and studied by Li and Richard [33] who call them 2-branch split disjunctions. A linear inequality valid for $P^{LP} \cap C_k(\pi_1, \pi_2, \gamma_1, \gamma_2)$ for $k = 1, \ldots, 4$, is called a *cross cut* for P obtained from the disjunction $C(\pi_1, \pi_2, \gamma_1, \gamma_2)$. As $P \subseteq \mathbb{Z}^{n_1} \times \mathbb{R}^{n_2} \subseteq C(\pi_1, \pi_2, \gamma_1, \gamma_2)$, cross cuts are valid for all points in P.

We define $P_C(\pi_1, \pi_2, \gamma_1, \gamma_2)$ as the convex hull of $P^{LP} \cap C(\pi_1, \pi_2, \gamma_1, \gamma_2)$. The *cross closure* of P, denoted by P_C, is the set of points in P^{LP} that satisfy all cross cuts obtained from all possible disjunctions for P. Clearly,

$$P_C = \bigcap_{\pi_1, \pi_2 \in \mathbb{Z}^{n_1}} \bigcap_{\gamma_1, \gamma_2 \in \mathbb{Z}} P_C(\pi_1, \pi_2, \gamma_1, \gamma_2).$$

4

2.3 Crooked cross disjunctions and crooked cross cuts

Similar to cross disjunctions, the *crooked cross disjunction* (see [18]) associated to $\pi_1, \pi_2 \in \mathbb{Z}^{n_1} \setminus \{\mathbf{0}\}$, and $\gamma_1, \gamma_2 \in \mathbb{Z}$ is given by

$$D(\pi_1, \pi_2, \gamma_1, \gamma_2) = \bigcup_{k \in \{1,2,3,4\}} D_k(\pi_1, \pi_2, \gamma_1, \gamma_2)$$

where:

$$
\begin{aligned}
D_1(\pi_1, \pi_2, \gamma_1, \gamma_2) &= \{(x, y) \in \mathbb{R}^{n_1+n_2} \ : \ \pi_1 x \leq \gamma_1, \ (\pi_2 - \pi_1)x \leq \gamma_2 - \gamma_1\}, \\
D_2(\pi_1, \pi_2, \gamma_1, \gamma_2) &= \{(x, y) \in \mathbb{R}^{n_1+n_2} \ : \ \pi_1 x \leq \gamma_1, \ (\pi_2 - \pi_1)x \geq \gamma_2 - \gamma_1 + 1\}, \\
D_3(\pi_1, \pi_2, \gamma_1, \gamma_2) &= \{(x, y) \in \mathbb{R}^{n_1+n_2} \ : \ \pi_1 x \geq \gamma_1 + 1, \ \pi_2 x \leq \gamma_2\}, \text{ and} \\
D_4(\pi_1, \pi_2, \gamma_1, \gamma_2) &= \{(x, y) \in \mathbb{R}^{n_1+n_2} \ : \ \pi_1 x \geq \gamma_1 + 1, \ \pi_2 x \geq \gamma_2 + 1\}.
\end{aligned}
$$

Notice that D_3 and D_4 above are same as C_3 and C_4 described in Section 2.2 whereas D_1 and D_2 are different from C_1 and C_2. A linear inequality valid for $P^{LP} \cap D_k(\pi_1, \pi_2, \gamma_1, \gamma_2)$ for $k = 1, \ldots, 4$, is called a *crooked cross (CC) cut* for P obtained from the disjunction $D(\pi_1, \pi_2, \gamma_1, \gamma_2)$. Clearly CC cuts are valid for all points in P.

We define $P_{CC}(\pi_1, \pi_2, \gamma_1, \gamma_2)$ to be the convex hull of $P^{LP} \cap D(\pi_1, \pi_2, \gamma_1, \gamma_2)$, and denote the *CC closure* of P by P_{CC} where

$$P_{CC} = \bigcap_{\pi_1, \pi_2 \in \mathbb{Z}^{n_1}} \bigcap_{\gamma_1, \gamma_2 \in \mathbb{Z}} P_{CC}(\pi_1, \pi_2, \gamma_1, \gamma_2).$$

It is easy to see that the CC closure and cross closure of P are contained in its split closure. Further, it is shown in [19] that $P_{CC} \subseteq P_C$, though it is not known if the containment is strict.

2.4 Breaking symmetry

We next discuss how to represent the disjunctions discussed above uniquely and how to identify "useless" disjunctions. First note that given $\pi_1, \pi_2 \in \mathbb{Z}^{n_1}$ and $\gamma_1, \gamma_2 \in \mathbb{Z}$, such that $S(\pi_1, \gamma_1) \subset S(\pi_2, \gamma_2)$, any split cut generated from $S(\pi_2, \gamma_2)$ can also be generated from $S(\pi_1, \gamma_1)$. This basic idea easily extends to more general disjunctions. We say that a split (or, cross, or crooked cross) disjunction "dominates" a second split (or, cross, or crooked cross) disjunction one if the first one is strictly contained in the second.

A split disjunction $S(\pi, \gamma)$ is dominated by another split disjunction unless components of π are coprime. In other words, $S(\pi, \gamma)$ is dominated if there exists an integer $k > 1$ such that $\pi/k \in \mathbb{Z}^{n_1}$; in this case $S(\pi/k, \lfloor \gamma/k \rfloor)$ is strictly contained in $S(\pi, \gamma)$. Conversely, as $S(\pi, \gamma)$ is dominated by $S(\pi', \gamma')$ only when the hyperplanes defined by π and π' are parallel to each other, it is easy to show that $S(\pi, \gamma)$ is not contained in another split disjunction if the components of π are coprime. Consequently, it suffices to only consider π that have coprime elements in (2).

It follows that for $\pi \neq \pi'$, $S(\pi, \gamma) = S(\pi', \gamma')$ if and only $\pi' = -\pi$ and $\gamma' = -\gamma - 1$. Notice that exactly one of γ and $-\gamma - 1$ is nonnegative and the other one is strictly negative. Consequently, in the definition of the split closure in (2), one can assume $\gamma \in \mathbb{Z}_+$. We therefore have the following observation.

Observation 2.1. *A split disjunction $S(\pi, \gamma)$ is uniquely defined by its parameters if $\gamma \geq 0$ and, it is not dominated by another split disjunction if and only if elements of π are coprime.*

It is relatively straight forward to extend this observation to cross disjunctions as follows after noticing the fact that $C(\pi_1, \pi_2, \gamma_1, \gamma_2) = C(\pi_2, \pi_1, \gamma_2, \gamma_1)$.

Observation 2.2. *A cross disjunction $C(\pi_1, \pi_2, \gamma_1, \gamma_2)$ is uniquely defined by its parameters if $\gamma_1, \gamma_2 \geq 0$ and, $[\pi_1, \gamma_1]$ is greater than $[\pi_2, \gamma_2]$ in the lexicographical sense. Furthermore, it is not dominated by another cross disjunction if elements of $[\pi_1, \pi_2]$ are coprime.*

When the elements of the vector $[\pi_1, \pi_2] \in \mathbb{Z}^{2n_1}$ are divisible by a common integer $k > 1$, then it is easy to show that $D(\pi_1, \pi_2, \gamma_1, \gamma_2)$ is dominated by $D((1/k)\pi_1, (1/k)\pi_2, \lfloor \gamma_1/k \rfloor, \lfloor \gamma_2/k \rfloor)$. However, due to the asymmetry in the description of crooked cross disjunctions, $D(\pi_1, \pi_2, \gamma_1, \gamma_2) = D(\pi_1', \pi_2', \gamma_1', \gamma_2')$ only when $\pi_1 = \pi_1', \pi_2 = \pi_2', \gamma_1 = \gamma_1'$ and $\gamma_2 = \gamma_2'$ and consequently these disjunctions are uniquely defined by the associated parameters. To see this, consider the crooked cross disjunction $D([1,0], [0,1], 0, 0)$ for $n_1 = 2$ as shown in Figure 1. Assume that there exists $\pi_1, \pi_2 \in \mathbb{Z}^2$ and $\gamma_1, \gamma_2 \in \mathbb{Z}$ such that $D(\pi_1, \pi_2, \gamma_1, \gamma_2) = D([1,0], [0,1], 0, 0)$. It is easy to see that either $\pi_1 = [1,0], \gamma_1 = 0$, or, $\pi_1 = [-1,0]$, $\gamma_1 = -1$. Furthermore, if $\pi_1 = [1,0]$ clearly $\pi_2 = [0,1]$ and $(\pi_1, \pi_2, \gamma_1, \gamma_2) = ([1,0], [0,1], 0, 0)$. On the other hand, when $\pi_1 = [-1,0]$ and $\gamma_1 = -1$, it is possible to show that there does not exist π_2, γ_2 such that $D(\pi_1, \pi_2, \gamma_1, \gamma_2) = D([1,0], [0,1], 0, 0)$ holds. We therefore make the following observation.

Observation 2.3. *A crooked cross disjunction $D(\pi_1, \pi_2, \gamma_1, \gamma_2)$ is uniquely defined by its parameters. Furthermore, it is not dominated by another crooked cross disjunction if elements of the extended vector $[\pi_1, \pi_2]$ are coprime.*
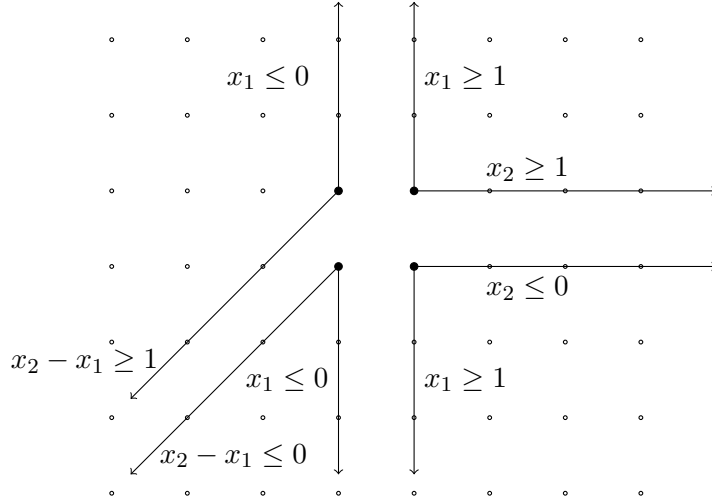

Figure 1: A crooked cross disjunction defined by $\pi_1 = [1,0], \gamma_1 = 0$ and $\pi_2 = [0,1], \gamma_2 = 0$

Given a pair of split disjunctions $S(\pi_1, \gamma_1)$ and $S(\pi_2, \gamma_2)$, we get only one "natural" cross disjunction by intersecting these two split disjunctions, which is $C(\pi_1, \pi_2, \gamma_1, \gamma_2)$. However, we can construct eight "natural" crooked cross disjunctions; we get different crooked cross disjunctions by switching (π_1, γ_1) and (π_2, γ_2), and by separately multiplying these vectors by ± 1.

3 Separating cross and crooked cross cuts

Given a mixed integer set $P = \{(x, y) \in \mathbb{Z}^{n_1} \times \mathbb{R}^{n_2} : Ax + Gy = b, x \geq 0, y \geq 0\}$, and a point $(\bar{x}, \bar{y}) \in P^{LP}$, it is easy to write a linear program to separate $(\bar{x}, \bar{y})$ from $P_C(\pi_1, \pi_2, \gamma_1, \gamma_2)$ if $\pi_1, \pi_2 \in \mathbb{Z}^{n_1}$,

and $\gamma_1, \gamma_2 \in \mathbb{Z}$ are fixed. More precisely, a violated cross cut of the form $ax + cy \geq d$ (here a and c are row vectors and d is a number), if it exists, can be obtained by solving the following linear program.

Cross Cut Separation LP:

$$\min \quad z = a\bar{x} + c\bar{y} - d \tag{6}$$

subject to

$$a \geq \lambda_1 A - \alpha_1 \pi_1 - \beta_1 \pi_2, \quad d \leq \lambda_1 b - \alpha_1 \gamma_1 - \beta_1 \gamma_2 \tag{7}$$

$$a \geq \lambda_2 A - \alpha_2 \pi_1 + \beta_2 \pi_2, \quad d \leq \lambda_2 b - \alpha_2 \gamma_1 + \beta_2 (\gamma_2 + 1) \tag{8}$$

$$a \geq \lambda_3 A + \alpha_3 \pi_1 - \beta_3 \pi_2, \quad d \leq \lambda_3 b + \alpha_3 (\gamma_1 + 1) - \beta_3 \gamma_2 \tag{9}$$

$$a \geq \lambda_4 A + \alpha_4 \pi_1 + \beta_4 \pi_2, \quad d \leq \lambda_4 b + \alpha_4 (\gamma_1 + 1) + \beta_4 (\gamma_2 + 1) \tag{10}$$

$$c \geq \lambda_i G \qquad\qquad \forall i \in \{1, 2, 3, 4\} \tag{11}$$

$$\beta_i, \alpha_i \geq 0, \qquad\qquad \forall i \in \{1, 2, 3, 4\} \tag{12}$$

$$\lambda_i \text{ free}, \qquad\qquad \forall i \in \{1, 2, 3, 4\} \tag{13}$$

$$a, c, d \text{ free}. \tag{14}$$

$$\sum_{i=1}^{4} (\|\lambda_i\|_1 + \alpha_i + \beta_i) \leq \quad 1 + n_1 + n_2. \tag{15}$$

Here $\lambda_1, \ldots, \lambda_4$ are row vectors with m components, and α_i and β_i are real numbers.

As $\mathbf{0}$ is a feasible solution to the cross cut separation LP, the optimal value $z^* \leq 0$. Note that if there exists a solution to the separation LP with $z < 0$, then the cross cut $ax + cy \geq d$ associated with this solution is violated by $(\bar{x}, \bar{y})$ and conversely, if there exists a violated cross cut, then there is a corresponding solution to the LP with $z < 0$. This implies that the optimal value $z^* = 0$ if and only if $(\bar{x}, \bar{y}) \in conv\left(P^{LP} \cap C(\pi_1, \pi_2, \gamma_1, \gamma_2)\right)$.

Observe that the constraints (7)-(14) define a cone, and thus the objective function value of the cross cut separation LP without the *normalization constraint* (15) is unbounded (when a violated cut exists). The normalization constraint makes the problem bounded while preserving all valid cuts up to scalar multiplication (see [32, 4]). Many other normalization constraints can be found in the literature, see [4]. We chose the one above based on some preliminary experiments.

3.1   Separating split cuts

Split cuts derived from a fixed split disjunction $S(\pi, \gamma)$ can be separated in a similar fashion by solving the *split cut separation LP* which we define by replacing the constraints (7)-(10) by

$$a \geq \lambda_1 A - \alpha_1 \pi, \quad d \leq \lambda_1 b - \alpha_1 \gamma$$
$$a \geq \lambda_2 A + \alpha_2 \pi, \quad d \leq \lambda_2 b + \alpha_2 (\gamma + 1)$$

and the number 4 by 2 in the constraints (11)-(13) and (15).

3.2   Separating crooked cross cuts

For a given fixed crooked cross disjunction $D(\pi_1, \pi_2, \gamma_1, \gamma_2)$, the associated separation problem can again be formulated as a linear program. The resulting LP is identical to (6)-(14) except constraints (7) and (8),

associated with the first two disjunctions are replaced with

$$a \geq \lambda_1 A - \alpha_1 \pi_1 - \beta_1(\pi_2 - \pi_1), \qquad d \leq \lambda_1 b - \alpha_1 \gamma_1 - \beta_1(\gamma_2 - \gamma_1) \qquad (16)$$
$$a \geq \lambda_2 A - \alpha_2 \pi_1 + \beta_2(\pi_2 - \pi_1), \qquad d \leq \lambda_2 b - \alpha_2 \gamma_1 + \beta_2(\gamma_2 - \gamma_1 + 1). \qquad (17)$$

The resulting LP produces a violated cut provided that there is one.

3.3 A bilinear integer program for separating from the (crooked) cross cut closure

If we let $\pi_1, \pi_2, \gamma_1, \gamma_2$ be variables in (6)-(14) we obtain a bilinear mixed integer separation problem for cross cuts given by (6)- (15), and

$$\pi_1, \pi_2 \in \mathbb{Z}^{n_1}, \ \gamma_1, \gamma_2 \in \mathbb{Z}. \qquad (18)$$

This separation problem will automatically select a cross disjunction from all possible cross disjunctions. A similar bilinear separation problem can be formulated for crooked cross cuts as well. Unfortunately, in both cases the bilinear problem is significantly harder to solve than the original separation problem as it contains integer variables as well as non-convex constraints. Still, the same can be said of the bilinear program for split cuts introduced in [23, 6] and these later formulations proved useful when (approximately) solved with specialized techniques. Developing similar techniques for the bilinear mixed integer separation problem for the cross closure or crooked cross closure is possible but it is beyond the scope of this paper.

3.4 When the separation LP fails

For a given disjunction, if the associated separation LP fails to produce a cut violated by the point $\bar{p} = (\bar{x}, \bar{y})$, useful information can be extracted from the optimal solution of the dual of the LP. For simplicity, consider a split disjunction $S(\pi, \gamma) = S_1(\pi, \gamma) \cup S_2(\pi, \gamma)$ such that $\bar{p} \notin S(\pi, \gamma)$ and assume that $\bar{p} \in P_S(\pi, \gamma)$. In this case, for some $1 > \mu > 0$

$$\bar{p} = \mu p^1 + (1 - \mu)p^2 \quad \text{such that} \quad p^i \in P^{LP} \cap S_i(\pi, \gamma), \ \text{for} \ i = 1, 2.$$

We refer to points p^1, p^2 as a pair of *friends* of $\bar{p}$. From a computational point of view, this information can be very useful when selecting a new disjunction from among a list of candidate disjunctions to find cuts separating the same point $\bar{p}$. More precisely consider a second disjunction $S(\pi', \gamma')$ not containing $\bar{p}$. If both p^1, p^2 obtained from the previous disjunction belong to $S(\pi', \gamma')$, then clearly $\bar{p} \in P_S(\pi', \gamma')$, and therefore no violated split cut can be derived from $S(\pi', \gamma')$. We discuss how we exploit this idea further in Section 6.
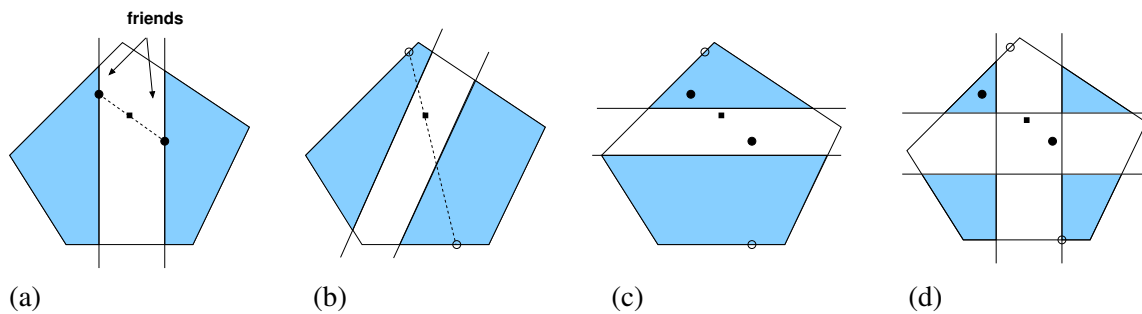

Figure 2: Friends of a point

The idea of friends is illustrated in Figure 2. For simplicity, this figure shows the projection onto a two dimensional space that contains the π's for all the split disjunctions considered. In Figure 2(a), we depict a pair of friends of $\bar{p}$ obtained from the displayed disjunction, say $S(\pi, \gamma)$. The polygon represents P^{LP}, the pair of parallel vertical lines represent the hyperplanes $\pi x = \gamma$ and $\pi x = \gamma + 1$ that define the disjunction and the shaded regions represent the atoms of the disjunction intersected with P^{LP}. The point $\bar{p}$ is represented by a filled square, and the friends are represented by filled circles. In Figure 2(b), we depict a pair of "good" friends of $\bar{p}$ (defined below) obtained from a different disjunction. In Figure 2(c), we depict a disjunction by a pair of horizontal lines, and show the two pairs of friends of $\bar{p}$ from Figures 2(b),(c). One pair does not belong to the disjunction, but the other pair does and consequently no violated split cut can be obtained from this disjunction. Figure 2(d) shows a cross disjunction that excludes one friend from each disjunction and therefore has the potential of producing violated cuts.

We next discuss how to generate a pair of friends from the separation LP when it does not produce a violated cut. Remember that the normalization constraint (15) requires taking the absolute value of the λ variables and therefore in the reformulation of the split cut separation LP below, we use two sets of nonnegative variables. Let $(\bar{x}, \bar{y}) \in P^{LP}$ denote the point to be separated.

$$\min \quad z = a\bar{x} + c\bar{y} - d \tag{19}$$

subject to

$$a \geq \lambda^{1,+} A - \lambda^{1,-} A - \alpha_1 \pi, \qquad d \leq \lambda^{1,+} b - \lambda^{1,-} b - \alpha_1 \gamma \tag{20}$$

$$a \geq \lambda^{2,+} A - \lambda^{2,-} A + \alpha_2 \pi, \qquad d \leq \lambda^{2,+} b - \lambda^{2,-} b + \alpha_2 (\gamma + 1) \tag{21}$$

$$c \geq \lambda^{i,+} G - \lambda^{i,+} G \qquad \text{for } i = \{1, 2\} \tag{22}$$

$$\lambda^{i,+}, \lambda^{i,-}, \beta_i, \alpha_i \geq 0, \qquad \text{for } i = \{1, 2\} \tag{23}$$

$$a, c, d \text{ free.} \tag{24}$$

$$\sum_{i=1}^{2} \left(\alpha_i + \beta_i + \sum_{j=1}^{m} (\lambda_j^{i,+} + \lambda_j^{i,-}) \right) \leq 1 + n_1 + n_2. \tag{25}$$

Note that when $(\bar{x}, \bar{y}) \in P_S(\pi, \gamma)$, the optimal solution to the separation LP has $z = 0$. Now consider the dual of this LP which also has an optimal value of 0.

$$\max \quad (1 + n_1 + n_2) \eta \tag{26}$$

subject to

$$Au^i + Gv^i \leq \omega_i b - \mathbf{1}\eta \qquad \text{for } i = 1, 2 \tag{27}$$

$$-Au^i - Gv^i \leq -\omega_i b - \mathbf{1}\eta \qquad \text{for } i = 1, 2 \tag{28}$$

$$-\pi u^1 \leq -\omega_1 \gamma - \eta \tag{29}$$

$$\pi u^2 \leq \omega_2 (\gamma + 1) - \eta \tag{30}$$

$$-u^1 - u^2 = \bar{x} \tag{31}$$

$$-v^1 - v^2 = \bar{y} \tag{32}$$

$$\omega_1 + \omega_2 = -1 \tag{33}$$

$$u^i, v^i, \omega_i, \eta \leq 0 \qquad \text{for } i = 1, 2 \tag{34}$$

Here, for $i = 1, 2$, $u^i \in \mathbb{R}^{n_1}$, $v^i \in \mathbb{R}^{n_2}$, and $\eta, \omega_i \in \mathbb{R}$.

Let $(\hat{u}^1, \hat{u}^2, \hat{v}^1, \hat{v}^2, \hat{\omega}^1, \hat{\omega}^2, \hat{\eta})$ be an optimal dual solution. As the optimal objective value of the dual problem equals 0, we have $0 = (1 + n_1 + n_2)\hat{\eta}$ and therefore $\hat{\eta} = 0$. Therefore, constraints (27) and (28) imply that $A\hat{u}^i + G\hat{v}^i = \omega_i b$ for $i = 1, 2$. First assume that $\hat{\omega}_1, \hat{\omega}_2 < 0$, and let

$$\bar{u}^i = \hat{u}^i/\hat{\omega}_i \quad \text{and} \quad \bar{v}^i = \hat{v}^i/\hat{\omega}_i$$

Note that $(\bar{u}^i, \bar{v}^i) \geq 0$ and $A\bar{u}^i + G\bar{v}^i = b$ and therefore $(\bar{u}^i, \bar{v}^i) \in P^{LP}$ for $i = 1, 2$. Next, notice that $(\bar{x}, \bar{y})$ is a convex combination of $(\bar{u}^1, \bar{v}^1)$ and $(\bar{u}^2, \bar{v}^2)$ as (31)–(34) imply that

$$\bar{x} = (-\hat{\omega}_1)\bar{u}^1 + (-\hat{\omega}_2)\bar{u}^2 \quad \text{and} \quad \bar{y} = (-\hat{\omega}_1)\bar{v}^1 + (-\hat{\omega}_2)\bar{v}^2. \tag{35}$$

Finally, due to (29) and (30), we have $(\bar{u}^i, \bar{v}^i) \in S_i(\pi, \gamma)$ and consequently, the points $(\bar{u}^i, \bar{v}^i)$ for $i = 1, 2$ give a pair of friends for $(\bar{x}, \bar{y})$.

Next consider the case when one of $\hat{\omega}_1, \hat{\omega}_2$ is zero. Without loss of generality, let $\hat{\omega}_1 = -1$ and $\hat{\omega}_2 = 0$. Let $\bar{u} = -\hat{u}^1$, $\bar{v} = -\hat{v}^1$ and $d_u = -\hat{u}^2$, $d_v = -\hat{v}^2$ and notice that by (27), (28) and (34), $(\bar{u}, \bar{v}) \in P^{LP}$ and (d_u, d_v) belongs to the recession cone of P^{LP}. Also notice that $\pi\bar{u} \leq \gamma$ and $\pi d_u \geq 0$ by (29) and (30), respectively. Furthermore, as $\bar{x} = \bar{u} + d_u$ by (31), and as $\pi\bar{x} > \gamma$ we have $\pi d_u > 0$. Therefore, $\pi(\bar{u} + \alpha d_u) \geq \gamma + 1$ for some $\alpha > 0$ and points $(\bar{u}, \bar{v})$ and $(\bar{u}, \bar{v}) + \alpha(d_u, d_v)$ give a pair of friends for $(\bar{x}, \bar{y})$.

Therefore if the split cut separation LP fails to produce a violated cut, then it is possible to produce a pair of friends of $\bar{p}$, namely, $p^1, p^2 \in P^{LP}$ such that $\bar{p} = (1 - \mu)p^1 + \mu p^2$ for some μ with $0 < \mu < 1$. In other words,

$$p^2 = p^1 + \frac{1}{\mu}(\bar{p} - p^1).$$

Now consider a point $p' = p^1 + \theta(\bar{p} - p^1)$ such that $\theta > 1/\mu$. If $p' \in P^{LP}$, then clearly p^1 and p' also form a pair of friends for $\bar{p}$. Moreover, from a computational point of view, this new pair is more useful than p^1, p^2 for checking if disjunctions other than $S(\pi, \gamma)$ have the potential to yield violated cuts. Assume $\bar{p} \notin S(\pi', \gamma') \neq S(\pi, \gamma)$. If $p^1, p^2 \in S(\pi', \gamma')$ (implying that $\bar{p} \in P_S(\pi', \gamma')$), then $p' \in S(\pi', \gamma')$. Consequently, it is best to find the point $\bar{p}^2 = p^1 + \theta(\bar{p} - p^1) \in P^{LP}$ such that θ is maximized. Similarly, one can find the point $\bar{p}^1 = p^2 + \theta(\bar{p} - p^2) \in P^{LP}$ such that θ is maximized. We call the new pair of friends $\bar{p}^1$ and $\bar{p}^2$ *good* friends of $\bar{p}$; see Figure 2(b).

It is possible to extend these ideas to separation LPs other than the split cut separation LP but the analysis is more tedious and as we discuss later in Section 6, we implemented this idea only for split cut separation.

4   Separation Heuristics

In this section we describe our separation heuristics to find violated cross cuts. Solving the separation problem exactly for split cuts is hard for practical instances [6, 23]. Therefore, we expect that finding violated cross or crooked cross cuts will also be very hard. For computational tractability, we do not attempt to generate a cross disjunction from scratch. Our simplest heuristic starts with a pair of given split disjunctions and just solves the cross cut separation LP to find a violated cut. Other heuristics start with a given split disjunction and then generate a second disjunction that leads to a violated cross cut. To obtain a list of potentially useful split disjunctions that we can use in these heuristics, we first generate violated split cuts and record the associated disjunctions. Assuming one of the disjunctions is fixed, we next discuss two ways of generating violated cross cuts.

10

In a recent paper, Dash and Goycoolea [20] describe a fast heuristic that finds violated split cuts using simplex tableau rows associated with feasible as well as infeasible bases. Given a point to be separated, their heuristic constructs a (possibly infeasible) basis and selects violated rank-1 GMI cuts from the corresponding tableau rows which, as discussed earlier, are split cuts. Their computational results show this approach produces cuts that give a reasonably good approximation of the split closure for problems in MIPLIB 3.0. Subsequent papers by Fischetti and Salvagnin [30] and Bonami [12] confirm this observation and give faster heuristics to find such cuts. Consequently, we assume that one can produce good split cuts for problems in MIPLIB 3.0 fairly quickly. The idea that violated split cuts and the associated split disjunctions can be found quickly is important to both our heuristics.

In the first heuristic, we collect a number of split cuts associated with a single split disjunction. In other words, we start off with a relaxation Q of $P_S(\pi, \gamma)$ for a fixed $\pi \in \mathbb{Z}^{n_1}$, and $\gamma \in \mathbb{Z}$. We then simply use the code of Dash and Goycoolea to generate rank-1 GMI cuts for Q. Clearly the generated cuts have split rank at most 2. We show below that they are also cross cuts. In the second heuristic, we set up an MIP to find a subclass of violated split cuts for $P_S(\pi, \gamma)$.

4.1 Rank 2 split cuts that are also cross cuts

We next describe a special family of rank-2 split cuts which are also cross cuts. As we discuss later, it is easy to show that the 2-step MIR inequalities [22] belong to this family of cuts. Note that no inclusion relationship between rank-2 split cuts and cross cuts is known, and in particular it is known that some cross cuts have unbounded split rank. For fixed $\pi \in \mathbb{Z}^{n_1}$ and $\gamma \in \mathbb{Z}$, remember that $P_S(\pi, \gamma)$ denotes the points in P^{LP} that satisfy all split cuts generated from the split disjunction $S(\pi, \gamma)$. We next present a basic observation which we later use to generate cross cuts.

Proposition 4.1. *Let $\pi_1 \in \mathbb{Z}^{n_1}$ and $\gamma_1 \in \mathbb{Z}$. Any split cut for $P_S(\pi_1, \gamma_1)$ is a cross cut for P. Furthermore, if a split cut for $P_S(\pi_1, \gamma_1)$ is obtained from the split disjunction $S(\pi_2, \gamma_2)$, then the same cut is a cross cut for P obtained from the cross disjunction $C(\pi_1, \pi_2, \gamma_1, \gamma_2)$.*

Proof. Let $ax + cy \geq d$ be a split cut for $P_S(\pi_1, \gamma_1)$ obtained from the split disjunction $S(\pi_2, \gamma_2)$. We will call $ax + cy \geq d$ simply "the cut" in the rest of the proof. By definition, the cut is valid for both $P_S(\pi_1, \gamma_1) \cap S_1(\pi_2, \gamma_2)$ and $P_S(\pi_1, \gamma_1) \cap S_2(\pi_2, \gamma_2)$ where $S_1(\pi_2, \gamma_2)$ and $S_2(\pi_2, \gamma_2)$ are the two half-spaces that define the disjunction $S(\pi_2, \gamma_2)$ as defined in Section 2.1. In addition, note that

$$P^{LP} \cap S_1(\pi_1, \gamma_1), \ P^{LP} \cap S_2(\pi_1, \gamma_1) \subseteq P_S(\pi_1, \gamma_1)$$

as $P_S(\pi_1, \gamma_1)$ is the convex hull of the union of these two sets.

Clearly, as the cut is valid for $P_S(\pi_1, \gamma_1) \cap S_1(\pi_2, \gamma_2)$, it is valid for its subsets

$$P^{LP} \cap S_1(\pi_1, \gamma_1) \cap S_1(\pi_2, \gamma_2) \text{ and } P^{LP} \cap S_2(\pi_1, \gamma_1) \cap S_1(\pi_2, \gamma_2).$$

Similarly, the cut is also valid for

$$P^{LP} \cap S_1(\pi_1, \gamma_1) \cap S_2(\pi_2, \gamma_2) \text{ and } P^{LP} \cap S_2(\pi_1, \gamma_1) \cap S_2(\pi_2, \gamma_2).$$

To conclude the proof, it is sufficient to observe that

$$S_i(\pi_1, \gamma_1) \cap S_j(\pi_2, \gamma_2) = C_{2i+j-2}(\pi_1, \pi_2, \gamma_1, \gamma_2)$$

for $i, j \in \{1, 2\}$, which shows that the cut is valid for $P^{LP} \cap C_k(\pi_1, \pi_2, \gamma_1, \gamma_2)$ for $k \in \{1, 2, 3, 4\}$. $\qquad \square$
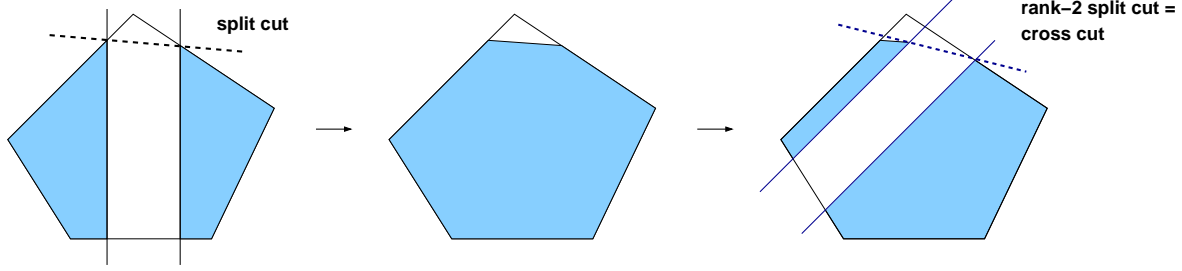
Figure 3: Rank-2 split cuts which are also cross cuts

In Figure 3 we show an example demonstrating how Proposition 4.1 works. The first picture shows $P^{LP} \cap S_1(\pi_1, \gamma_1)$ and $P^{LP} \cap S_2(\pi_1, \gamma_1)$, and the second picture shows their convex hull $P_S(\pi_1, \gamma_1)$. The last picture shows a split cut for $P_S(\pi_1, \gamma_1)$ which is also a cross cut.

We next describe the so-called 2-step MIR inequalities and show that they are cross cuts. Dash and Günlük [22] study the following simple mixed-integer set

$$Q = \{y \in \mathbb{R},\ x_1, x_2 \in \mathbb{Z}\ :\ y + \alpha x_1 + x_2\ \geq\ \beta,\ \ y, x_1 \geq 0\}.$$

where $\beta, \alpha \in \mathbb{R}$ and $1 > \beta > \alpha > 0$, and show that

$$\big(1/(\beta - \alpha \lfloor \beta/\alpha \rfloor)\big)y + x_1 + \lceil \beta/\alpha \rceil x_2\ \geq\ \lceil \beta/\alpha \rceil \tag{36}$$

is valid (and facet defining) for Q provided that $1/\alpha \geq \lceil \beta/\alpha \rceil > \beta/\alpha$. The validity proof in [22] essentially shows that the 2-step MIR inequality (36) is an MIR inequality (or, split cut) for the set $Q' \subseteq Q$ obtained by strengthening the original set with the simple MIR inequality

$$y + \alpha x_1 + \beta x_2\ \geq\ \beta,$$

obtained using the disjunction $\{x\ :\ x_2 \leq 0\} \cup \{x\ :\ x_2 \geq 1\}$. As inequality (36) is a split cut for $Q' \supseteq Q_S([0,1],0)$, by Proposition 4.1, it is a cross cut for the original set Q.

4.2   An MIP based heuristic when one of the disjunctions is fixed

As discussed in Section 2.1, given $(\bar{x}, \bar{y}) \in P^{LP}$, finding a violated split cut is equivalent to finding an implied equation of the form (3), such that the associated MIR cut (4) is violated by $(\bar{x}, \bar{y})$. This is a difficult computational task. Notice that when $\hat{\beta} \neq 0$, the MIR cut (4) can also be rewritten as

$$\sum_{i=1}^{n_1} (\lfloor a_i \rfloor + \min(\frac{\hat{a}_i}{\hat{\beta}}, 1)) x_i + \frac{1}{\hat{\beta}} \sum_{i=1}^{n_2} \max(c_i, 0) y_i \geq \lceil \beta \rceil.$$

Therefore, for any positive $\epsilon \leq \hat{\beta}$, the following *approximate MIR* cut

$$\sum_{i=1}^{n_1} \lceil a_i \rceil x_i + \frac{1}{\varepsilon} \sum_{i=1}^{n_2} \max(c_i, 0) y_i \geq \lceil \beta \rceil, \tag{37}$$

is a weakening of (4) and hence it is also a split cut obtained from the disjunction (5). Also notice that inequality (37) becomes the so-called pro-CG cut (see [13]) if all c_i are non-positive. It is possible to write

an MIP model for separating this approximate MIR cut and therefore separating approximate MIR cuts is easier than separating actual MIR cuts. Clearly, the existence of a violated MIR cut does not guarantee the existence of a violated approximated MIR cut. However, the MIP model would always return a base inequality from which an MIR cut can be derived. Additionally, the associated split disjunction can be used to obtain other split cuts using a separation LP (we note that this is the underlying idea used in [6] to separate split cuts).

We next combine the idea of separating approximate MIR cuts with Proposition 4.1 to obtain a separation MIP for cross cuts. In this model $\epsilon \in (0,1)$ is a given constant and $a, \tilde{a}, \tilde{c}$ are unknown row vectors with n_1, n_1 and n_2 components respectively.

$$\min \quad z = \tilde{a}\bar{x} + \frac{1}{\varepsilon}\tilde{c}\bar{y} - (h+1) \tag{38}$$

subject to

$$a \geq \lambda_1 A + \alpha_1 \pi, \qquad h + \varepsilon \leq \lambda_1 b + \alpha_1 \gamma \tag{39}$$

$$a \geq \lambda_2 A - \alpha_2 \pi, \quad h + \varepsilon \leq \lambda_2 b - \alpha_2(\gamma + 1) \tag{40}$$

$$\tilde{c} \geq \lambda_i G \text{ for } i = 1,2 \qquad\qquad \tilde{c} \geq 0 \tag{41}$$

$$\alpha_i \geq 0 \text{ for } i = 1,2 \tag{42}$$

$$\tilde{a} \geq a \tag{43}$$

$$\tilde{a} \in \mathbb{Z}^{n_1}, h \in \mathbb{Z} \tag{44}$$

$$\tilde{a}, a, h, \lambda_1, \lambda_2 \text{ free.} \tag{45}$$

Consider a feasible solution for this model. Let $c \in \mathbb{R}^{n_2}$ be a vector defined by $c = \max\{\lambda_1 G, \lambda_2 G\}$ with the maximum being taken componentwise. Then $ax + cy \geq h + \varepsilon$ is a valid inequality for $P^{LP} \cap S_1(\pi, \gamma)$ and also for $P^{LP} \cap S_2(\pi, \gamma)$. Further,

$$\tilde{a}x + \frac{1}{\varepsilon}\tilde{c}y \geq h + 1 \tag{46}$$

is an approximate MIR cut derived from $ax + cy \geq h + \varepsilon$ using the disjunction, say $S(\pi', \gamma')$, in (5) with β standing for $h + \varepsilon$. Therefore (46) is a cross cut for the cross disjunction $C(\pi, \pi', \gamma, \gamma')$. Additionally, even if the objective of this model is nonpositive we can still recover $S(\pi', \gamma')$ and use the separation LP for $C(\pi, \pi', \gamma, \gamma')$ to obtain additional cross cuts.

5 Cross cuts from two tableau rows

In this section we report on our computational experiments with cross cuts that can be obtained from two-row relaxations of the initial MIP formulation. We do not consider crooked cross cuts as our preliminary computational experiments suggest that the additional gap closed by crooked cross cuts over cross cuts is not significant; we quantify this more precisely in Section 6. Our aim here is to establish that cuts obtained from two row relaxations are indeed stronger than cuts that are obtained from one row relaxations. On the MIPLIB 3.0 problem set we observe that cuts from two rows increase the lower bound significantly.

5.1 One-row vs two-row relaxations

Given a mixed-integer program of the form

$$\min\{c^T x : Ax = b, \ l \leq x \leq u, \ x_k \in \mathbb{Z} \ \forall k \in S\},$$

where S contains the indices of the integer variables, it is possible to obtain a reformulation by using an optimal basis of the LP-relaxation:

$$\min\{c^T x : x_B + A_B^{-1} A_N x_N = A_B^{-1} b, \ l \le x \le u, \ x_k \in \mathbb{Z} \ \forall k \in S\},$$

where x_B and x_N denote the basic and non-basic variables and A_B and A_N denote the submatrix of A corresponding to the the basic and non-basic variables, respectively. Using this reformulation, one can define one-row relaxations of the feasible region:

$$P_i = \{x : x_{B_i} + \sum_{k \in N} \bar{a}_{ik} x_k = \bar{b}_i, \ l \le x \le u, \ x_k \in \mathbb{Z} \ \forall k \in S\},$$

one for each basic, integral variable x_{B_i} ($i \in \{1, \ldots, m\}, B_i \in S$) where $\bar{b}_i$ is fractional.

It is well-known that the Gomory mixed-integer (GMI) cut derived from the ith row of the simplex tableau is a valid inequality for P_i. Following Balas et. al. [5], we refer to the step of generating one GMI cut from each P_i and simultaneously adding all violated ones as a *round* of GMI cuts. In [5], the authors demonstrate that adding all violated GMI cuts is more effective than adding a fraction of the violated cuts.

It has been computationally established for a number of practical MIP instances that adding one round of GMI cuts has the same effect as adding all possible inequalities that are valid for some P_i. Subsequent to the work in [21] and [30] comparing the effectiveness of one round of GMI cuts to different classes of "one-row" cuts, Fukasawa and Goycoolea [31] showed that for most MIPLIB 3.0 problems adding all (knapsack) cuts based on the relaxations P_i does not yield improved bounds over that obtained by adding one round of GMI cuts. Thus, they essentially show that no additional cuts from individual rows of an optimal simplex tableau are useful over and above GMI cuts from these rows for some practical problems.

Therefore, a natural question is whether one can obtain useful valid inequalities by using pairs of simplex tableau rows that give relaxations of the form:

$$P_{ij} = \{x : x_{B_i} + \sum_{k \in N} \bar{a}_{ik} x_k = \bar{b}_i, \ x_{B_j} + \sum_{k \in N} \bar{a}_{jk} x_k = \bar{b}_j, \ l \le x \le u, \ x_k \in \mathbb{Z} \ \forall k \in S\},$$

where $i, j \in \{1, \ldots, m\}$, $B_i, B_j \in S$, and both $\bar{b}_i$ and $\bar{b}_j$ are fractional. Clearly $P_{ij} = P_i \cap P_j$. In this section we do not attempt to generate all valid inequalities for P_{ij} but instead, restrict our attention to those that can be generated as cross cuts.

5.2 Computational approach and separation

Given a mixed-integer program, our first step is to relax integrality and solve the LP relaxation to obtain an optimal solution and basis. Using this optimal solution, we first identify the rows of the tableau that lead to violated GMI cuts (i.e., the rows i such that $B_i \in S$ and $\bar{b}_i$ is fractional). To avoid numerical difficulties, we ignore tableau rows which have $\max\{|a_{ik}|/|a_{il}| : k, l \in N\} > 10^9$. We then construct the two-row relaxations P_{ij} for pairs of tableau rows such that (i) the GMI cuts derived from both P_i and P_j are violated by the initial LP solution, (ii) the defining equations for P_i and P_j have common non-basic variables with non-zero coefficients, and (iii) $i > j$ (to avoid repetition). Notice that we ignore all pairs i, j where the associated tableau rows do not have any variables in common. The motivation for this is that both split cuts from such P_{ij} are implied by split cuts (and cross cuts) from P_i and P_j, and the experimental work by Fukasawa and Goycoolea [31] suggests that cuts in addition to the GMI cuts from individual rows are unlikely to be useful. In our experiments, we observe that the number of pairs to be considered often

drops significantly leading to much faster computation times, without a noticeable decrease in final bounds obtained.

The following is a summary of our cross cut separation algorithm:

Cross cut separation from pairs of tableau rows:

Solve P^{LP} and for selected tableau rows, construct P_{ij} from all row pairs.
Add one round of GMI cuts to P^{LP} to obtain $\bar{P}$, resolve, and let $\bar{x}$ be its solution.
Repeat
    for all P_{ij}
        Generate a list of cross cuts for P_{ij};
        if ($\bar{x}$ violates some of the cuts) then
            update $\bar{P}, \bar{x}$ by adding violated cuts to $\bar{P}$;
            exit the for loop;
        end if
    end for
Until (no violated cuts are found by the for loop above)

It is clear that we may consider a relaxation P_{ij} for some fixed i and j many times in order to generate cuts, which differs from the approach described in [24] and [8]; we also do not restrict the number of cuts added.

For a given P_{ij}, we generate a list of cross cuts using the ideas presented in Section 4.1. Let P_{ij}^i be obtained by augmenting P_{ij} by the GMI cut derived from P_i and define P_{ij}^j similarly. After creating the sets P_{ij}^i and P_{ij}^j we invoke the rank-1 GMI heuristic of Dash and Goycoolea [20] for each of them, in particular, by invoking the FEAS, SPARSE and RANDOM heuristics; see [20, Table 4]. By the discussion in Section 4.1, the resulting rank-2 split cuts are also cross cuts for P_{ij}. We refer to this separation method as Cross.def. In an alternate implementation, we invoke the cross cut separation LP in Section 3 with the input problem being P_{ij}, and the cross disjunction being defined by the split disjunctions associated with the GMI cuts from P_i and P_j. This separation method is called Cross.LP1.

Note that when we obtain a rank-1 GMI cut for P_{ij}^i, the associated split disjunction (say S_i') can be different from the split disjunction associated with the GMI cut from P_i (say S_i) or from P_j (say S_j). Further, this rank-1 GMI cut is a cross cut obtained from the pair of disjunctions S_i and S_i'. We save S_i' in a list associated with S_i and in a separation method called Cross.LP2, we solve the cross cut separation LP for P_{ij} with the input cross disjunction defined by the split disjunctions S_i and S_i'. The motivation for doing so is that not all cross cuts from a cross disjunction can be obtained as rank-2 split cuts. It is well-known that a type 1 triangle cut is a cross cut but has infinite split rank.

Our final way of obtaining cross cuts from P_{ij} is via the MIP based heuristic described in Section 4.2 which starts with a fixed disjunction and finds a second one to generate a violated cut. We refer to the associated algorithm as Cross.IP.

We also separate split cuts from the same collection of two-row relaxations P_{ij}. Our separation procedure for split cuts is same as the one we use for cross cuts except the word "cross" in the fifth line is replaced by the word "split", and we apply the rank-1 GMI heuristic above to P_{ij}. We call this algorithm Split.sep. As we discuss later, we observe that split cuts from two-row relaxations yield significantly better bounds than one round of GMI cuts. Clearly the bound after adding cross cuts should be better than that from split cuts and this is indeed the case, though not by a very large amount.

5.3 Computational Environment

Our computational results are obtained on a 2.93 GHz Intel Xeon machine running the Linux operating system. We solve linear programs and auxiliary integer programs (in Cross.IP) with IBM ILOG CPLEX 12.2. We do not report our computing times for the algorithms Cross.LP1, Cross.LP2, and Cross.IP; our implementation of these routines is not competitive with the code for Split.sep and Cross.def.

5.4 Computational Experiments

In Table 1, we discuss the integrality gap closed with two combinations of algorithms for 54 out of 65 instances from MIPLIB 3.0. The discarded instances have no integrality gap after GMI cuts are added, or no integrality gap is closed after extensive generation of split cuts as reported in [6, 23]); they are *dsbmip, enigma, noswot, air03, 10teams, mod010, markshare1, markshare2, pk1, stein27*, and *stein45*. Further, we replace free variables in any remaining instance with the difference of two nonnegative variables. The problem name is given in the column titled "problem". The next two columns give the average gap closed by one round of GMI cuts, and the number of violated GMI cuts. In the subsequent two columns we give the gap closed by invoking Cross.def, and the number of cuts added. We only invoke Cross.def after first invoking Split.sep. In the last two columns (under the heading ALL), we give the gap closed and cuts added by invoking all of the separation algorithms discussed above. We invoke them in the following order: Split.sep, Cross.def, Cross.LP2, Cross.LP1, Cross.IP.

Note that for many problems we obtain a significant increase over the gap closed by one round of GMI cuts by adding a small number (about the same order of magnitude as the number of GMI cuts) of cuts generated from two tableau rows with Cross.def. The numbers for the `gesa` problems, for example, are especially striking, as the gap closed increases by more than 30% with the addition of a small number of cuts. The additional gap closed as we move from Cross.def to ALL is not as striking, as can be seen in the averages in Table 2.

In Table 2, we compare the average gap closed (in the second column) for the 54 MIPLIB 3.0 problems in our testbed via the different cuts and separation methods (in the first column). In the fourth row, we show that the gap closed with Cross.def is 37.10% as opposed to 25.80% with GMI cuts alone across all problems, and is thus a nontrivial improvement. However, in the third row, we see that split cuts from pairs of tableau rows provide a good amount of this improvement. From the fourth row onwards we add separation methods one at a time on top of the previous ones to measure their marginal impact. The fifth row indicates whether the disjunctions from which we obtain our cross cuts as rank-2 split cuts in Cross.def can yield much stronger cross cuts. Clearly this is not the case as the gap closed increases from 37.10% to 37.63%. This is not shocking as the relaxations P_{ij} have only two rows, and we would expect a few cuts to give a good approximation to the integer hull. Observe that the gap closed by Cross.LP1 is much less than Cross.def (and Split.sep); this means that the disjunctions found by Cross.def are useful, and the the disjunctions associated with the first round of GMI cuts are not enough to improve the bounds a lot. On the other hand, these disjunctions improve the gap closed by Cross.def + Cross.LP2 to 38.05% in the sixth row. Finally, Cross.IP obtains additional useful disjunction and improve the gap closed to 38.39%.

We note that the average gap closed for 53 out of these 54 problems considered in Louveaux and Poirrier [34] (they have numerical difficulties with *dano3mip*) is 32.38%. For these problems, the respective values for GMI + Split.sep and Cross.Def are 36.44% and 37.80%. However, the experiments are not comparable, as Louveaux and Poirrier do not fix the tableau but update it up to five times, and also consider only a subset

problem	GMI		Cross.def		ALL		problem	GMI		Cross.def		ALL	
	% gap	cuts	% gap	cuts	% gap	cuts		% gap	cuts	% gap	cuts	% gap	cuts
air04	6.71	292	12.71	195	12.71	195	misc03	7.24	18	10.34	12	11.55	17
air05	4.64	224	8.80	154	8.80	154	misc06	29.40	13	30.77	2	31.31	11
arki001	29.26	56	46.77	99	55.13	580	misc07	0.72	26	0.72	12	0.72	12
bell3a	60.43	32	67.15	3	67.15	3	mitre	80.76	796	88.64	115	88.64	115
bell5	14.53	25	17.41	4	17.78	12	mkc	1.21	70	23.72	241	27.11	822
blend2	16.36	6	18.07	23	21.62	58	mod008	20.89	5	61.61	58	65.56	106
cap6000	41.65	2	62.73	28	63.76	144	mod011	17.11	16	17.20	6	18.73	73
dano3mip	0.02	97	0.03	15	0.03	15	modglob	15.10	30	35.33	82	35.44	105
danoint	1.74	52	1.74	12	1.74	97	nw04	62.27	6	68.44	6	68.92	7
dcmulti	43.08	49	52.21	43	54.62	83	p0033	54.60	6	64.91	19	66.80	49
egout	55.93	40	58.45	7	58.85	9	p0201	18.24	20	24.42	7	24.42	14
fast0507	1.68	306	1.89	3	1.89	4	p0282	3.70	26	33.54	213	40.68	533
fiber	65.02	41	76.49	27	78.87	226	p0548	39.46	47	68.70	461	73.49	3971
fixnet6	10.87	60	11.51	32	11.71	57	p2756	0.46	36	0.54	13	2.63	81
flugpl	11.74	10	13.70	9	14.10	13	pp08a	52.88	51	68.84	44	70.28	57
gen	61.62	43	79.10	22	79.10	22	pp08aCUTS	30.07	41	41.75	48	42.47	64
gesa2	27.19	58	64.32	42	65.00	68	qiu	1.99	36	1.99	1	2.60	46
gesa2_o	30.21	73	63.10	42	63.20	69	qnet1	12.73	49	28.92	142	29.40	321
gesa3	45.87	85	82.69	36	83.01	44	qnet1_o	30.71	11	34.43	61	36.49	284
gesa3_o	50.57	100	85.25	30	85.32	35	rentacar	29.05	16	29.05	0	29.15	2
gt2	67.72	11	69.90	47	77.09	3314	rgn	4.49	16	24.67	45	27.92	80
harp2	8.69	10	13.70	68	14.78	628	rout	0.32	29	4.01	169	4.03	504
khb05250	74.91	19	88.06	18	88.56	20	set1ch	38.11	138	60.86	218	60.86	218
l152lav	1.55	51	26.08	122	26.09	124	seymour	8.39	598	10.11	12	10.11	12
lseu	48.42	12	57.23	34	59.81	321	swath	17.66	45	33.38	10	33.38	10
mas74	6.67	12	9.46	101	9.82	156	vpm1	9.45	15	11.15	6	11.15	6
mas76	6.42	11	10.26	68	11.83	476	vpm2	12.58	30	26.50	51	26.63	64

Table 1: gap closed with two row cuts on MIPLIB problems

of all tableau row pairs.

In Table 3, we give the average gap closed (in the second column) and geometric means of computations times in the third column for different variants of Cross.def. The time for an algorithm is the sum of the time to find cuts and the time to reoptimize after adding cuts. First notice that Split.sep takes about 100 times the computing time of a round of GMI cuts. For only about 1.34% extra gap closed, Cross.def takes more than twice the time as Split.sep.

It is natural to ask whether we can obtain the gap closed by Cross.def by choosing only a subset of GMI cuts, and generating cuts from associated pairs of rows, or by carefully choosing pairs of rows. After all, in Cross.def, often many pairs of rows have to be examined before cuts are found. We attempt to answer this question by the computational experiments summarized in Table 4. In all algorithms considered in this experiment (except in Cross.def.nosort), each time we start the for loop in Section 5.2, we first sort the GMI cuts in decreasing order of dual values assigned to these cuts in the previous (strengthened) LP relaxation. Our purpose here is to distinguish between more and less important GMI cuts. For example, Cross.def x 30% means that we generate cross cuts from pairs of rows associated with the top 30% of GMI cuts only. The next two rows have a similar meaning. In addition, in the for loop, the sets P_{ij} are considered in decreasing order of sums of dual values for the associated GMI cuts. For example, the first set

Cut Separation Method	gap closed
GMI	25.80
GMI + Cross.LP1	32.70
GMI + Split.sep	35.76
Previous + Cross.def	37.10
Previous + Cross.LP2	37.63
Previous + Cross.LP1	38.05
Previous + Cross.IP	38.39

Table 2: Comparison of different algorithms

Cut Separation Method	gap closed	time
GMI	25.80	0.02
Split.sep	35.76	1.90
Cross.def	37.10	4.26

Table 3: Comparison of different methods

P_{ij} considered for cross cut separation would be the set associated with the two rows that give the two "most important" GMI cuts (provided that they have common non-basic variables). The time taken by Cross.def x 30% is only 10 times the time to compute GMI cuts, but it does not close too much more gap. As more pairs of rows are considered for cross cut separation, more of the gap is closed but at a cost of higher computation time. Cross.def.nosort means we do not sort the GMI cuts by dual values, and simply consider them in the order they appear in the tableau and a pair $(i_1, j_1) < (i_2, j_2)$ if the first pair is lexicographically less then the second one. It does seem to take noticeably more time than Cross.def to close essentially the same gap (this relationship also holds between the sorted and non-sorted variants of Cross.def x 30% etc.). Thus our heuristic to order pairs seems useful, but our heuristic to drop pairs altogether seems less so. Finally, we can get the same gap as Cross.def, but in less time if we only consider pairs of tableau rows which have common variables that are basic in the current relaxation; this is given in Cross.def.basici.

Cut Separation Method	gap closed	time
Cross.def	37.10	4.26
Cross.def x 30%	29.47	0.11
Cross.def x 50%	32.31	0.47
Cross.def x 70%	34.56	1.57
Cross.def.nosort	36.81	6.08
Cross.def.basici	36.94	3.65

Table 4: Comparison of different variants of Cross.def

6 Bounds on cross and crooked cross cut closures

In this section we report on our computational experiments with cross and crooked cross cuts that are obtained using the full formulation of a MIP as opposed to using only two-row relaxations. Our aim in this

section is to establish that cross and crooked cross cuts give better lower bounds than split cuts on practical problem instances, namely on the MIPLIB 3.0 problem set. In earlier experiments with split cuts, Balas and Saxena [6] and Dash, Günlük, and Lodi [23] present separation models that, at least in principle, can optimize over the split closure to any degree of precision. We do not propose any such mechanism for the cross or crooked cross cut closure; instead, we use a collection of heuristics to obtain effective cuts.

We present two computational experiments below. The first experiment compares the effect of split cuts with the effect of cross and crooked cross cuts for a given list of split disjunctions. To this end, we construct a (short) list of split disjunctions and compare the lower bound obtained after adding all possible split cuts from this list with the lower bound obtained after adding all possible cross and crooked cross cuts generated using pairs of split disjunctions from the same list. In the second experiment, we use cross cut heuristics which generate new cross disjunctions. With our heuristics, we are able to obtain better bounds than the best known split closure bounds for a number of problem instances suggesting that cross cuts are potentially valuable for solving mixed-integer programs.

6.1   Experiments with GMI disjunctions

In our first experiment, we obtain all violated GMI cuts (which are also split cuts) from the optimal simplex tableau together with the split disjunctions associated with them, which we refer to as the GMI disjunctions. (As in the previous section, we ingore tableau rows where the ratio of two coefficients is too large.) Let these disjunctions be $S(\pi^i, \gamma^i)$ for $i \in I$. Let z^S be the lower bound obtained after adding all split cuts from the GMI disjunctions. We compare the associated integrality gap closed with the gap closed after adding the GMI cuts, and with the gap closed by all cross cuts that can be derived from pairs of GMI disjunctions.

To compute z^S, we first solve P^{LP}, add all violated GMI cuts, and obtain the collection of disjunctions I associated with these cuts. We then repeatedly execute the following pseudocode until no more violated split cuts can be found.

Split cut separation from GMI disjunctions:
  for $i \in I$
      Solve the split cut separation LP for $S(\pi^i, \gamma^i)$;
      if (a violated cut is found) then add it to the cut pool;
      if (the cut pool contains 10 cuts or more) then exit the for loop;
  end for
  add all cuts in the cut pool to the current LP relaxation and resolve;

Similarly, let z^C denote the lower bound obtained after adding all cross cuts obtained from pairs of GMI disjunctions. We compute z^C by repeatedly executing the following pseudocode until no more violated cuts can be found.

Cross cut separation from GMI disjunctions:
  Separate all split cuts from GMI disjunctions;
  for $i, j \in I$
      Solve the cross cut separation LP for $C(\pi^i, \pi^j, \gamma^i, \gamma^j)$;
      if (a violated cut is found) then add it to the cut pool;
      if (the cut pool contains 5 cuts or more) then exit the for loop;
  end for
  add all cuts in the cut pool to the current LP relaxation and resolve;

Cuts separated	% gap closed
GMI cuts	26.25
Split cuts from GMI disjunctions	41.42
Cross cuts from GMI disjunctions	43.96
Crooked cross cuts from GMI disjunctions	45.15
Best split closure bound	79.89

Table 5: Experiments with GMI disjunctions on MIPLIB problems

Notice that we start separating cross cuts only after all violated split cuts are added to the formulation. Also note that once some violated cross cuts are found, we again look for violated split cuts before we proceed with cross cut separation. In other words, to speed up the computation, we go back to finding violated split cuts whenever a small number of cross cuts is found. Finally, we find all possible crooked cross cuts from every pair of GMI disjunctions, using a code similar to the one above for cross cuts, except that we have to consider eight different crooked cross disjunctions (and associated LPs) for any pair of split disjunctions. We use the ideas described in Section 3.4 to avoid solving LPs whenever possible and thus speed up computation. We next briefly describe how, without solving the LP, we decide that the separation LP for a class of cuts cannot yield a violated cut.

Consider the split cut separation algorithm described above and notice that a point $\bar{p}$ violates a split cut from the disjunction $S(\pi^i, \gamma^i)$ only if $\bar{p} \notin S(\pi^i, \gamma^i)$. In other words, if the point satisfies the disjunction, then it satisfies all the split cuts that can be generated using that disjunction and therefore there is no need to solve the split cut separation LP. Furthermore, as discussed in Section 3.4, for each disjunction $S(\pi^i, \gamma^i)$ for which we have solved the split cut separation LP and failed to find a violated cut, we obtain a pair (q_1^i, q_2^i) of (good) friends of $\bar{p}$. When we consider a subsequent split disjunctions to separate $\bar{p}$, we know that only disjunctions that do not contain $\bar{p}$ and at least one of q_1^i, q_2^i have the potential to produce violated cuts. Similarly, when we separate cross cuts (and this happens only after all split cut separation LPs fail to produce a violated cut) we use the list of friends generated by the split cut separation LPs. We solve a separation LP for the cross disjunction $C(\pi^i, \pi^j, \gamma^i, \gamma^j)$ for some $i, j \in I$ only when it does not contain $\bar{p}$ and it does not contain at least one of q_1^k and q_2^k for each pair of friends (q_1^k, q_2^k) for $k \in I$. With this idea we are able to check a large number of cross disjunctions for the potential to generate violated cuts in reasonable time. We use the same idea for testing crooked cross disjunctions.

In Table 5 we report on the average integrality gap closed in this fashion for 53 out of the 65 MIPLIB 3.0 problems. We consider the same 54 instances discussed in the previous section, but discard *fast0507* here (and in all other experiments in this section) as we have numerical difficulties with this instance. The rows of the table from the second one onwards give the gap closed by, respectively, the first round of GMI cuts, split cuts from the GMI disjunctions, cross cuts from the GMI disjunctions, crooked cross cuts, and the best known split closure bounds from [6, 23, 20].

We make a few immediate observations based on Table 5. The list of disjunctions associated with the first round of GMI cuts does not yield a good approximation of the split closure on the average. However, separating additional split cuts from these disjunctions can significantly improve the bound obtained by one round of GMI cuts. Cross cuts based on pairs of these disjunctions do yield a non-trivial yet not very substantial improvement in the bound from split cuts. The improvement in bound due to crooked cross cuts is a bit misleading as all the improvement is due to the extra gap closed for two problems, namely *gesa2* (about 29%) and *modglob* (about 16%). Without these two instances, the extra gap closed is very small on

the average, and we therefore do not try to separate crooked cross cuts in subsequent experiments.

The bound obtained with cross cuts is still very far from the best split closure bound. Note that this also means that we cannot conclude that the cross cuts from these disjunctions have split rank larger than 1 with respect to P (of course, we would need additional disjunctions to derive these cross cuts as split cuts for P).

6.2 Experiments with cross cuts from heuristic disjunctions

In this section we describe our efforts to exceed the best known elementary split closure bounds using cross cuts. As reported by various authors (see [6, 23, 20]), the elementary split closure provides very good bounds for MIPs in the MIPLIB 3.0 library. Therefore, it is desirable to start with a collection of split cuts (and the associated disjunctions) that give a reasonable approximation of the split closure bounds and then generate cross cuts. To this end, we use a slightly modified version of the code of Dash and Goycoolea [20] that generates rank-1 split cuts by heuristically generating (possibly infeasible) bases of the LP relaxation, and the associated tableau rows and GMI cuts. We use their default heuristic DEF, and we modify their code to store the split disjunctions whenever a violated cut is found. We will call this heuristic the DG heuristic from now on. In addition, to control computation time, we terminate the DG heuristic when the number of generated cuts (and therefore disjunctions) exceeds 1000.

In [20], the split cuts generated by the heuristic DEF close 61.90% of the integrality gap for the 53 MIPLIB 3.0 problems presented in Table 5; if we terminate the DG heuristic after 1000 cuts, we get a weaker bound in some cases. Compared to the best known gap closed by all split cuts, which is 79.89%, this is a reasonable number as it sometimes takes many hours to obtain the split closure bounds, see [6].

Let $\{S(\pi^i, \gamma^i) : \text{ for } i \in \bar{I}\}$ be the list of split disjunctions associated with the split cuts returned by the DG heuristic; this collection has up to 1000 disjunctions. We then run the split cut separation algorithm described in Section 6.1 with this list of disjunctions instead of the GMI disjunctions only. This gives a reasonably good bound compared with the split closure bounds. While generating split cuts, we keep track of which cuts are generated using which disjunction. More precisely, for disjunction $S(\pi^i, \gamma^i)$ for $i \in \bar{I}$, we create a list of cuts K_i generated using that disjunction. Unlike our experiments with the GMI disjunctions discussed earlier, we do not proceed to the cross cut separation via LPs right away with pairs of split disjunctions associated with $\bar{I}$. Instead, we proceed with the separation heuristics described in Sections 4.1 and 4.2. To find rank-2 split cuts that are also cross cuts we use the following heuristic repeatedly until no cuts can be found.

Cross cut separation using rank-2 split cuts:
    for $i \in \bar{I}$
        Obtain $ApxP_S(\pi^i, \gamma^i)$ by adding K_i to P^{LP};
        Use the DG heuristic to find violated rank-1 split cuts for $ApxP_S(\pi^i, \gamma^i)$;
        if (violated cuts are found) then
            add the associated disjunctions to D_i;
            add the cuts to the current LP relaxation and resolve;
            exit the for loop;
        end if
    end for

When separating cross cuts using this heuristic, notice that we also build a list of split disjunctions D_i associated with each input disjunction $S(\pi^i, \gamma^i)$ for $i \in \bar{I}$. Remember that any violated split cut for $ApxP_S(\pi^i, \gamma^i)$ generated using the disjunction $S(\pi, \gamma) \in D_i$ is also a cross cut for P derived from the cross

disjunction $C(\pi_i, \pi, \gamma_i, \gamma)$. Therefore the pair of disjunctions $S(\pi^i, \gamma^i)$ and $S(\pi, \gamma)$ is a "good pair" for separating cross cuts. We implicitly save this pair and solve an associated cross cut separation LP later on as described below in order to get cross cuts not obtainable in line 3 of the heuristic above (recall that some cross cuts have infinite split rank).

Cross cut separation using "good pairs" of disjunctions:

   for $i \in \bar{I}$
       for $S(\pi, \gamma) \in D_i$
           Solve the cross cut separation LP for $C(\pi^i, \pi, \gamma^i, \gamma)$;
           if (a violated cut is found) then add it to the cut pool;
           if (the cut pool contains 5 cuts or more) then exit the outer for loop;
       end for
   end for
   add all cuts in the cut pool to the current LP relaxation and resolve;

    We use our MIP based separation heuristic to get cross cuts as follows.

Cross cut separation using MIP-based heuristic:

   for $i \in \bar{I}$
       Use the MIP heuristic to find the most violated approximate GMI cut for $P_S(\pi^i, \gamma^i)$;
       Obtain the associated split disjunction $S(\pi', \gamma')$;
       Solve the cross cut separation LP with the cross disjunction $C(\pi^i, \pi', \gamma^i, \gamma')$ as input;
       if (a violated cut is found) then
         add it to the the current LP relaxation and resolve;
         exit the for loop;
       end if
   end for

    Our final heuristic solves LPs to generate cross cuts from pairs of disjunctions associated with the list $\bar{I}$ in exactly the same way as the heuristic "Cross cut separation from GMI disjunctions" generates cuts from the list I. We combine the heuristics discussed in this section in the following manner:

1. Run the DG heuristic to obtain the list $\bar{I}$ with up to 1000 split cuts and disjunctions.

2. Separate split cuts via a separation LP for each $i \in \bar{I}$.

3. Separate rank-2 split cuts for each $i \in \bar{I}$; if successful go to Step 2.

4. Separate cross cuts using the MIP-based heuristic for each $i \in \bar{I}$; if successful go to Step 2.

5. Separate cross cuts from "good pairs" of disjunctions; if successful go to Step 2.

6. Separate cross cuts from all pairs of disjunctions associated with $\bar{I}$; if successful go to Step 2.

6.3   Computational results

It is known from earlier work that the split closure bound can be very strong for some of the MIPLB 3.0 problems. In our computational experiments we exclude all instances where at least 99% of the integrality gap is closed by split cuts. For these instances it is very hard to exceed the split closure bound and one can

| | | rank-1 | | rank-2 | | split | |
problem	GMI	GMI	split cuts	split cuts	cross cuts	closure	time (sec)
bell5	14.5	25.9	86.2	92.7	99.8	92.9	300
cap6000	41.7	62.4	63.6	65.3	66.1	65.2	+
gesa3	45.9	90.9	94.9	95.4	96.6	95.8	+
gesa3 o	50.6	91.3	95.2	95.5	97.4	95.2	+
gt2	67.7	96.6	96.7	98.3	99.2	98.4	200
mas76	6.4	10.7	18.8	19.3	29.1	26.5	9500
modglob	15.1	81.3	88.3	93.1	99.6	92.2	+
p0033	54.6	84.1	86.2	90.3	100.0	87.4	40
p0201	18.2	69.5	73.9	93.9	97.0	74.9	+
pp08a	52.9	94.1	95.1	97.0	97.8	97.0	+
qiu	2.0	21.8	78.1	78.1	78.1	77.5	+
rentacar*	29.1	39.9	46.9	46.9	46.9	44.9	+
set1ch	38.1	86.7	87.6	96.9	96.9	89.7	+
mkc*	1.2	4.1	51.5	51.5	51.5	49.3	+

Table 6: Gap closed with cross cuts on MIPLIB problems

argue that there is also not much point in attempting to do so. In addition, we also exclude all instances where the integrality gap closed via the best split closure bound is 1% or less. After this, we are left with 32 out of 65 MIPLIB 3.0 instances. We next report on the best bounds we obtained with our heuristics with a time limit of 10 hours. In Table 6, we report on 14 out of these 32 instances, where we exceed the best known split closure bound, taken from [6, 23, 20] (given in the seventh column), by at least 0.5%. In the first column, we give the problem name. In the second, third and forth columns, we give the bound obtained by GMI cuts from the initial tableau, split cuts using the DG heuristic and split cuts using the disjunctions found by the DG heuristic. The next two columns report on the bound obtained, respectively, by cross cuts which are rank-2 split cuts, and by all cross cuts. The last column reports on the overall computation time and the symbol "+" means that the code reached the time limit.

For three problems *qiu, rentacar* and *mkc*, we were not able to complete separating split cuts within the time limit and we report on better split closure bounds for these problems using our heuristics and simply repeat the same bound in the remaining columns. In the remaining 11 instances, the bound we obtain using split cuts is worse than the split closure bound. For example, for *mas76*, our initial list of split disjunctions closes less than 19% of the integrality gap, whereas the split closure bound is 26.5%. We are able to exceed the split closure bound for these instances using cross cuts. In particular, note that for the problem *p0033*, 87.4% of the integrality gap is closed by split cuts (and this number is close to the best split closure bound) and cross cuts close all the remaining integrality gap within 40 seconds. Also note that for most of the instances, the algorithm terminated due to the time limit of 10 hours.

Our results show that cross cuts can yield an improvement over split cuts for some MIPLIB problems. We also note that our procedure is computationally intensive and more work is needed to make it more efficient.

References

[1] K. Andersen, Q. Louveaux, R. Weismantel, and L. Wolsey, Cutting planes from two rows of a simplex tableau, Proceedings 12th Conference on Integer Programming and Combinatorial Optimization (LNCS 4513) (M. Fischetti and D. P. Williamson, eds.), Springer-Verlag, 2007, pp. 1–15.

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

[3] E. Balas, Intersection cuts from maximal lattice-free convex sets and lift-and-project cuts from multiple-term disjunctions, Presentation at the Spring Meeting of the American Mathematical Society (Western Section), San Francisco, 2009.

[4] E. Balas, P. Bonami, Generating Lift-and-Project Cut from the LP Simplex Tableau: Open Source Implementation and Testing of New Variants, *Mathematical Programming Computation* **1** (2009) 165–199.

[5] E. Balas, S. Ceria, G. Cornuejols, and N. Natraj, Gomory cuts revisited, *Operations Research Letters* **19** 1–9, 1996.

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

[7] A. Basu, P. Bonami, G. Cornuéjols, and F. Margot, On the relative strength of split, triangle and quadrilateral cuts. *Mathematical Programming*, to appear.

[8] A. Basu, P. Bonami, G. Cornuéjols and F. Margot, Experiments with Two-Row Cuts from Degenerate Tableaux, *INFORMS Journal on Computing*, to appear.

[9] A. Basu, M. Campelo, M. Conforti, G. Cornuéjols and G. Zambelli. On Lifting Integer Variables in Minimal Inequalities, IPCO 2010, LNCS 6080 (2010) 85–95.

[10] A. Basu, M. Conforti, G. Cornuéjols and G. Zambelli. Minimal inequalities for an infinite relaxation of integer programs, *SIAM Journal on Discrete Mathematics* **24** (2010) 158–168.

[11] 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.

[12] P. Bonami. On optimizing over lift-and-projetc closures, Manuscript, 2010.

[13] P. Bonami, G. Cornuéjols, S. Dash, M. Fischetti and A. Lodi. Projected Chvátal-Gomory cuts for mixed integer linear programs, *Mathematical Programming* **113** (2008) 241–257.

[14] V. Borozan and G. Cornuéjols, Minimal Valid Inequalities for Integer Constraints, *Mathematics of Operations Research* **34** (2009) 538–546.

[15] M. Conforti, G. Cornuéjols, G. Zambelli, A geometric perspective on lifting, *Operations Research*, to appear, 2009.

[16] M. Conforti, G. Cornuéjols, G. Zambelli, Corner Polyhedron and Intersection Cuts, *Surveys in Operations Research and Management Science* **16** (2011) 105–120.

[17] W. J. Cook, R. Kannan, and A. Schrijver, Chvátal closures for mixed integer programming problems *Mathematical Programming* **47** (1990) 155–174.

[18] S. Dash, S. S. Dey, and O. Günlük, Two dimensional lattice-free cuts and asymmetric disjunctions for mixed-integer polyhedra, *Mathematical Programming*, to appear, 2010.

[19] S. Dash, S. S. Dey, and O. Günlük, On mixed-integer sets with two integer variables, *Operations Research Letters*, to appear, 2011.

[20] S. Dash, M. Goycoolea, A heuristic to find violated rank-1 GMI cuts. *Mathematical Programming Computation* **2** (2010) 231–258.

[21] S. Dash and O. Günlük, On the strength of Gomory mixed-integer cuts as group cuts, *Mathematical Programming* **115** (2008) 387–407.

[22] S. Dash and O. Günlük, Valid Inequalities Based on Simple Mixed-integer Sets, *Mathematical Programming*, **105** (2006) 29–53.

[23] S. Dash, O. Günlük, and A. Lodi, MIR closures of polyhedral sets, *Mathematical Programming* **121**(1) (2010) 33–60.

[24] S. S. Dey, A. Lodi, A. Tramontani, and L. A. Wolsey, Experiments With Two Row Tableau Cuts, IPCO 2010.

[25] S. S. Dey and A. Tramontani, Recent developments in multi-row cuts, *Optima* **80** (2009) 2–8.

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

[27] S. S. Dey and L. A. Wolsey, Constrained infinite group relaxations of MIPs, *SIAM Journal on Optimization* **20** (2010) 2890–2912.

[28] D. Espinoza, Operations Research Letters **38** (2010) 115–120.

[29] M. Fischetti and C. Saturni, Mixed integer cuts from cyclic groups, *Mathematical Programming* **109** (2007) 27–53.

[30] M. Fischetti, and D. Salvagnin, A relax-and-cut framework for Gomory's mixed-integer cuts, *Mathematical Programming Computation*, to appear, 2011.

[31] R. Fukasawa and O. Günlük, Strengthening lattice-free cuts with nonnegativity, *Discrete Optimization*, **8** (2011) 229–245.

[32] M. Fischetti, A. Lodi, A. Tramontani, On the separation of disjunctive cuts, *Mathematical Programming* (2009), DOI:10.1007/s10107-009-0300-y.

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

[34] Q. Louveaux and L. Poirrier, An algorithm for the separation of two-row cuts, `http://www.optimization-online.org/DB__HTML/2011/03/2967.html`.