Math. Program., Ser. B 94: 221–245 (2003)

Digital Object Identifier (DOI) 10.1007/s10107-002-0317-y

Egon Balas · Michael Perregaard

A precise correspondence between lift-and-project cuts, simple disjunctive cuts, and mixed integer gomory cuts for 0-1 programming

Received: October 5, 2000 / Accepted: March 19, 2002
Published online: September 5, 2002 – © Springer-Verlag 2002

Abstract. We establish a precise correspondence between lift-and-project cuts for mixed 0-1 programs, simple disjunctive cuts (intersection cuts) and mixed-integer Gomory cuts. The correspondence maps members of one family onto members of the others. It also maps bases of the higher-dimensional cut generating linear program (CGLP) into bases of the linear programming relaxation. It provides new bounds on the number of facets of the elementary closure, and on the rank, of the standard linear programming relaxation of the mixed 0-1 polyhedron with respect to the above families of cutting planes.

Based on the above correspondence, we develop an algorithm that solves (CGLP) without explicitly constructing it, by mimicking the pivoting steps of the higher dimensional (CGLP) simplex tableau by certain pivoting steps in the lower dimensional (LP) simplex tableau. In particular, we show how to calculate the reduced costs of the big tableau from the entries of the small tableau and based on this, how to identify a pivot in the small tableau that corresponds to one or several improving pivots in the big tableau. The overall effect is a much improved lift-and-project cut generating procedure, which can also be interpreted as an algorithm for a systematic improvement of mixed integer Gomory cuts from the small tableau.

1. Introduction

Cutting planes for integer programs, pure or mixed, 0-1 or general, have a 40-some-years history. Gomory's mixed integer cuts were proposed in the early sixties [13]. In the late sixties, intersection cuts made their appearance [1]. Soon they developed into disjunctive cuts, of which a variety were proposed in the seventies. A subclass of the latter were revived in the early nineties under the name of lift-and-project cuts [4], and were implemented in branch and cut algorithms [5], a framework which proved to be particularly fruitful. Gomory's mixed integer cuts are also closely related to disjunctive cuts. Specifically, it has been shown (in e.g. [14]) that they are a special case of disjunctive cuts from a two-sided disjunction, also called split cuts [10].

Here we give a precise characterization of the connection between lift-and-project cuts and the earlier cuts in this literature. This correspondence has theoretical and practical consequences. On the theoretical side, it provides new bounds on the number of essential cuts in the elementary closure, and on the rank of the standard relaxation of a

E. Balas: Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA 15213-3890, USA; e-mail: eb17@andrew.cmu.edu

Research was supported by the National Science Foundation through grant #DMI-9802773 and by the Office of Naval Research through contract N00014-97-1-0196.

M. Perregaard: Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA 15213-3890, USA; e-mail: michael4@andrew.cmu.edu

mixed 0-1 polyhedron with respect to various families of cuts. On the practical side, it makes it possible to solve the cut generating linear program of the lift-and-project procedure on the simplex tableau of the standard LP relaxation, without explicit recourse to the expanded formulation. The algorithm that does this can also be interpreted as a procedure for systematically improving a mixed integer Gomory cut from the optimal simplex tableau through a sequence of pivots that combine the terms of the disjunction applied to the cut row with other rows of the same tableau in a specific way.

Next we outline the structure of the paper. The rest of this section states the problem. Section 2 describes simple disjunctive cuts and mixed integer Gomory cuts, while section 3 does the same thing for lift-and-project cuts. Section 4 establishes the precise correspondence between lift-and-project cuts and simple disjunctive cuts, while section 5 does the same for the strengthened version of these cuts, which includes the mixed integer Gomory cuts. Section 6 derives some bounds on the number of undominated disjunctive cuts, while section 7 gives bounds on the rank of the linear programming polyhedron with respect to various families of cuts. Section 8 describes the algorithm for solving the cut generating linear program implicitly, through appropriate pivots in the simplex tableau of the LP relaxation, while section 9 discusses preliminary computational experience with this approach as compared to solving the cut generating linear program directly in the higher dimensional space. Finally, section 10 interprets the algorithm of section 8 as a method for improving mixed integer Gomory cuts in a systematic fashion.

*

We consider the mixed integer 0-1 program in the form

$$\begin{aligned} \min\ & cx \\ s.t.\ & \\ & Ax \geq b \\ & x \geq 0 \\ & x_j \in \{0, 1\},\ j = 1, \dots, p \end{aligned} \tag{MIP}$$

where A is $(m + p) \times n$, c and x are n-vectors, and b is an $(m + p)$-vector for some p, $0 \leq p \leq n$. We will assume that the system $Ax \geq b$ subsumes the inequalities $x_j \leq 1$, $j = 1, \dots, p$, i.e. the last p inequalities of $Ax \geq b$ are $-x_j \geq -1$, $j = 1, \dots, p$. The linear programming relaxation of (MIP) is

$$\min\{cx : x \in P\}, \tag{LP}$$

where

$$P := \{x \in \mathbb{R}^n_+ : Ax \geq b\}.$$

We will sometimes denote the constraint set defining P by $\tilde{A}x \geq \tilde{b}$, where $\tilde{A} := \binom{A}{I}$ and $\tilde{b} := \binom{b}{0}$ have $m + p + n$ rows. The vector $\bar{x}$ will denote an optimal solution to (LP). To simplify notation we will often write x_J to mean the subvector of the components of x indexed by the index set J, and A^R_Q to mean the submatrix of A whose rows and columns are indexed by the index sets Q and R, respectively.

Let $S := \{1, \dots, m + p\}$ and $N := \{m + p + 1, \dots, m + p + n\}$ index the surplus variables and the structural variables in (LP), respectively, where we use s to denote the vector of surplus variables in $Ax \geq b$. With this indexing we have a direct correspondence between the variables of (LP) and the surplus variables of $\tilde{A}x \geq \tilde{B}$. Note, however, that the latter system contains the extra surplus variables from the rows

$$x_j - s_j = 0, \quad j = m + p + 1, \ldots, m + p + n \tag{1}$$

which of course are equal to the corresponding structural variables.

The simplex tableau for (LP) is determined uniquely by the set of variables chosen to be nonbasic. If we let I and J index the set of basic and nonbasic variables, respectively, then the simplex tableau for (LP) with such a choice of basis can be written

$$\begin{array}{l} x_i + \sum_{j \in N \cap J} \bar{a}_{ij} x_j + \sum_{j \in S \cap J} \bar{a}_{ij} s_j = \bar{a}_{i0} \text{ for } i \in N \cap I \\ s_i + \sum_{j \in N \cap J} \bar{a}_{ij} x_j + \sum_{j \in S \cap J} \bar{a}_{ij} s_j = \bar{a}_{i0} \text{ for } i \in S \cap I \end{array} \tag{2}$$

Here $\bar{a}_{ij}$ denotes the coefficient for nonbasic variable j in the row for basic variable i, and $\bar{a}_{i0}$ is the corresponding right-hand side constant. When dealing with a row of the simplex tableau (2) we will identify the nonbasic variables $x_{N \cap J}$ with the corresponding s_J so that we can write a row k of the tableau in the more conscise form

$$x_k + \sum_{j \in J} \bar{a}_{kj} s_j = \bar{a}_{k0}$$

2. Simple disjunctive cuts and mixed integer Gomory cuts

Consider the simplex tableau associated with the optimal solution $\bar{x}$ to (LP), and let the row associated with basic variable x_k be

$$x_k = \bar{a}_{k0} - \sum_{j \in J} \bar{a}_{kj} s_j \tag{3}$$

where J is the index set of nonbasic variables, and $0 < \bar{a}_{k0} < 1$. The *intersection cut* [1] from the convex set $\{x \in \mathbb{R}^n : 0 \leq x_k \leq 1\}$, also known as the *simple disjunctive cut* from the condition $x_k \leq 0 \vee x_k \geq 1$ applied to (3), is $\pi s_J \geq \pi_0$, where $\pi_0 := \bar{a}_{k0}(1 - \bar{a}_{k0})$ and

$$\pi_j := \max\{\bar{a}_{kj}(1 - \bar{a}_{k0}), -\bar{a}_{kj}\bar{a}_{k0}\} \quad j \in J. \tag{4}$$

Note that the cut $\pi s_J \geq \pi_0$ derived from $x_k \leq 0 \vee x_k \geq 1$ depends on the nonbasic set J in terms of which x_k is expressed. Different sets J give rise to different cuts derived from the same disjunction $x_k \leq 0 \vee x_k \geq 1$.

When $p \geq 1$, the simple disjunctive cut $\pi s_J \geq \pi_0$ can be strengthened [2, 6] by replacing π with $\bar{\pi}$, defined as

$$\bar{\pi}_j := \begin{cases} \min\{f_{kj}(1 - \bar{a}_{k0}), (1 - f_{kj})\bar{a}_{k0}\} & j \in J \cap \{1, \ldots, p\} \\ \pi_j \quad j \in J \setminus \{1, \ldots, p\} \end{cases} \tag{5}$$

with $f_{kj} := \bar{a}_{kj} - \lfloor \bar{a}_{kj} \rfloor$.

The strengthened simple disjunctive cut $\bar{\pi} s_J \geq \pi_0$ is the same as the mixed integer Gomory cut [13]. The mixed integer Gomory cut, when applied to a pure 0-1 program, dominates the fractional Gomory cut for pure integer programs [12].

3. Lift-and-project cuts

Lift-and-project cuts [4] are a special class of *disjunctive cuts* [2, 3], obtained from a disjunction of the form

$$\begin{pmatrix} Ax \geq b \\ x \geq 0 \\ -x_k \geq 0 \end{pmatrix} \vee \begin{pmatrix} Ax \geq b \\ x \geq 0 \\ x_k \geq 1 \end{pmatrix} \qquad (6)_k$$

for some $k \in \{1, \ldots, p\}$ such that $0 < \bar{x}_k < 1$. A lift-and-project cut $\alpha x \geq \beta$ from this disjunction is obtained by solving the cut generating linear program (see [4])

$$\begin{array}{llllll} \min \alpha\bar{x} - \beta & & & & & \\ \alpha & - & uA + u_0 e_k & & & \geq 0 \\ \alpha & & & - & vA - v_0 e_k & \geq 0 \\ & -\beta + & ub & & & = 0 \\ & -\beta & & + & vb + v_0 & = 0 \\ & & \sum_{i=1}^{m+p} u_i + u_0 & + & \sum_{i=1}^{m+p} v_i + v_0 & = 1 \\ & & u, u_0, v, v_0 \geq 0, & & & \end{array} \qquad (\text{CGLP})_k$$

where e_k is the k-th unit vector.

The last equation of $(\text{CGLP})_k$ is a normalization constraint, meant to truncate the polyhedral cone defined by the remaining inequalities. The objective function of $(\text{CGLP})_k$ is chosen so as to maximize the amount by which $\bar{x}$ is cut off (see [4]).

While an optimal solution to $(\text{CGLP})_k$ yields a deepest cut in this sense, any solution to the constraint set of $(\text{CGLP})_k$ yields a member of the family of lift-and-project cuts. However, since cuts corresponding to nonbasic solutions are dominated by those corresponding to basic solutions, we will only be interested in the latter.

Since the components of (α, β) are unconstrained in sign, they can be eliminated and $(\text{CGLP})_k$ can be solved solely in terms of the variables (u, u_0, v, v_0). Given any basic solution (u, u_0, v, v_0) to this reduced system, the (α, β) component of the corresponding basic solution to $(\text{CGLP})_k$, and hence the coefficient vector of the cut $\alpha x \geq \beta$, is defined by

$$\beta := ub = vb + v_0,$$

and

$$\alpha_j := \begin{cases} \max\{ua_j, va_j\} & j \neq k \\ \max\{ua_k - u_0, va_k + v_0\} & j = k, \end{cases} \qquad (7)$$

where a_j denotes the j-th column of A.

The lift-and-project cuts $\alpha x \geq \beta$ defined this way are derived from the disjunction on the 0-1 variable x_k considered in the form $(6)_k$. The integrality conditions on x_j, $j \in \{1, \ldots, p\} \setminus \{k\}$, can be used to strengthen these cuts [2,4,5], and α can be replaced by $\bar{\alpha}$, where

$$\bar{\alpha}_j := \begin{cases} \min\{ua_j + u_0\lceil m_j \rceil, va_j - v_0\lfloor m_j \rfloor\}, & j \in \{1, \ldots, p\} \setminus \{k\}, \\ \alpha_j, \quad j \in \{k\} \cup \{p+1, \ldots, n\}, \end{cases} \qquad (8)$$

with

$$m_j := \frac{va_j - ua_j}{u_0 + v_0}. \tag{9}$$

We will call $\alpha x \geq \beta$ defined by (7) an (unstrengthened) lift-and-project cut, and the cut $\bar{\alpha} x \geq \beta$ defined by (8), (9) a strengthened lift-and-project cut.

Next we establish a precise connection between the lift-and-project cuts (strengthened lift-and-project cuts) on the one hand, and simple disjunctive cuts (strengthened simple disjunctive cuts or mixed integer Gomory cuts) on the other. A first step in this direction was taken in [4] where it was shown how one can obtain a feasible solution to (CGLP) from the basis inverse corresponding to an optimal basic solution to (LP).

4. The correspondence between the unstrengthened cuts

First, we introduce surplus variables into the inequalities of $(\mathrm{CGLP})_k$ and rewrite the constraint set as

$$\begin{array}{llllll}
\alpha & - & u\tilde{A} + u_0 e_k & & & = 0 \\
\alpha & & & - & v\tilde{A} - v_0 e_k & = 0 \\
 & -\beta + & u\tilde{b} & & & = 0 \\
 & -\beta & & + & v\tilde{b} + v_0 & = 0 \\
 & & \sum_{i=1}^{m+p+n} u_i + u_0 & + & \sum_{i=1}^{m+p+n} v_i + v_0 & = 1 \\
 & & u, u_0, v, v_0, \geq 0, & & &
\end{array} \tag{10}$$

where the vectors u, v now have among their components the surplus variables.

Lemma 1. *In any basic solution to (10) that yields an inequality $\alpha x \geq \beta$ not dominated by the constraints of (LP), both u_0 and v_0 are positive.*

Proof. If $u_0 = 0$, then $\alpha = u\tilde{A}$, $\beta = u\tilde{b}$; and if $v_0 = 0$, then $\alpha = v\tilde{A}$, $\beta = v\tilde{b}$. In either case, $\alpha x \geq \beta$ is a nonnegative linear combination of the inequalities of $\tilde{A}x \geq \tilde{b}$. □

Since the components of (α, β) are unrestricted in sign, we may assume w.l.o.g. that they are all basic. We then have

Lemma 2. *Let $(\bar{\alpha}, \bar{\beta}, \bar{u}, \bar{u}_0, \bar{v}, \bar{v}_0)$ be a basic solution to (10), with $\bar{u}_0, \bar{v}_0 > 0$ and all components of $(\bar{\alpha}, \bar{\beta})$ basic. Further, let the basic components of $\bar{u}$ and $\bar{v}$ be indexed by M_1 and M_2, respectively. Then $M_1 \cap M_2 = \emptyset$, $|M_1 \cup M_2| = n$, and the $n \times n$ submatrix $\hat{A}$ of $\tilde{A}$ whose rows are indexed by $M_1 \cup M_2$ is nonsingular.*

Proof. Removing from (10) the nonbasic variables and subscripting the basic components of u and v by M_1 and M_2, respectively, we get

$$\begin{array}{llllll}
\alpha & - u_{M_1}\tilde{A}_{M_1} & + u_0 e_k & & & = 0 \\
\alpha & & & - v_{M_2}\tilde{A}_{M_2} & - v_0 e_k & = 0 \\
-\beta + & u_{M_1}\tilde{b}_{M_1} & & & & = 0 \\
-\beta & & & + v_{M_2}\tilde{b}_{M_2} & + v_0 & = 0 \\
 & u_{M_1}\mathbf{1}_{|M_1|} + & u_0 & + v_{M_2}\mathbf{1}_{|M_2|} & + v_0 & = 1,
\end{array} \tag{11}$$

where $\tilde{A}_{M_1}$ ($\tilde{A}_{M_2}$) is the submatrix of $\tilde{A}$ whose rows are indexed by M_1 (by M_2).

Eliminating the variables α, β, unrestricted in sign, we obtain the system

$$\begin{aligned} (u_{M_1}, -v_{M_2})\binom{\tilde{A}_{M_1}}{\tilde{A}_{M_2}} \quad - (u_0 + v_0)e_k &= 0 \\ (u_{M_1}, -v_{M_2})\binom{\tilde{b}_{M_1}}{\tilde{b}_{M_2}} \qquad\qquad - v_0 &= 0 \\ u_{M_1}\mathbf{1}_{|M_1|} + v_{M_2}\mathbf{1}_{|M_2|} + \; u_0 + v_0 &= 1 \end{aligned} \tag{12}$$

of $n+2$ equations, of which $(\bar{u}_{M_1}, \bar{u}_0, \bar{v}_{M_2}, \bar{v}_0)$ is the unique solution. Since the number of variables, like that of the constraints, is $n+2$, it follows that $|M_1| + |M_2| = n$.

Now suppose that

$$\hat{A} := \begin{pmatrix} \tilde{A}_{M_1} \\ \tilde{A}_{M_2} \end{pmatrix}$$

is singular. Then there exists a vector $(u^*_{M_1}, v^*_{M_2})$ such that $(u^*_{M_1}, -v^*_{M_2})\hat{A} = 0$ and $u^*_{M_1}\mathbf{1}_{|M_1|} + v^*_{M_2}\mathbf{1}_{|M_2|} = 1$. By setting

$$v^*_0 = (u^*_{M_1}, -v^*_{M_2})\begin{pmatrix} \tilde{b}_{M_1} \\ \tilde{b}_{M_2} \end{pmatrix} \quad , \quad u^*_0 = -v^*_0$$

we obtain a solution $(u^*_{M_1}, u^*_0, v^*_{M_2}, v^*_0)$ to (12). By assumption $\bar{u}_0 > 0$ and $\bar{v}_0 > 0$, hence the solution $(u^*_{M_1}, u^*_0, v^*_{M_2}, v^*_0)$ differs from $(\bar{u}_{M_1}, \bar{u}_0, \bar{v}_{M_2}, \bar{v}_0)$. But this contradicts that $(\bar{u}_{M_1}, \bar{u}_0, \bar{v}_{M_2}, \bar{v}_0)$ is the unique solution to (12), which proves that $\hat{A}$ is nonsingular.

If $|M_1 \cap M_2| \neq \emptyset$ then $\hat{A}$ will be singular, hence $|M_1 \cap M_2| = \emptyset$ and $|M_1 \cup M_2| = n$. $\square$

Now define $J := M_1 \cup M_2$, and consider the system obtained from $\tilde{A}x \geq \tilde{b}$ by replacing the n inequalities indexed by J with equalities, i.e. by setting the corresponding surplus variables to 0. Since the submatrix $\hat{A}$ of $\tilde{A}$ whose rows are indexed by J is nonsingular, these n equations define a basic solution, with an associated simplex tableau whose *nonbasic* variables are indexed by J. Recall that in the $(\text{CGLP})_k$ solution that served as our starting point, $J := M_1 \cup M_2$ was the index set of the *basic* components of (u, v). Writing $\hat{b}$ for the subvector of $\tilde{b}$ corresponding to $\hat{A}$, and s_J for the surplus variables indexed by J, we have

$$\hat{A}x - s_J = \hat{b}$$

or

$$x = \hat{A}^{-1}\hat{b} + \hat{A}^{-1}s_J. \tag{13}$$

Here some components of s_J may be surplus variables in an inequality of the form $x_j \geq 0$. Such a variable is of course equal to, and therefore can be replaced by, x_j itself. If that is done, then the row of (13) corresponding to x_k (a basic variable, since $k \notin J$) can be written as

$$x_k = \bar{a}_{k0} - \sum_{j \in J} \bar{a}_{kj} s_j \tag{14}$$

where $\bar{a}_{k0} = e_k\hat{A}^{-1}\hat{b}$ and $\bar{a}_{kj} = -(\hat{A}^{-1})_{kj}$. Notice that (14) is the same as (3).

Note that this basic solution (and the associated simplex tableau) need not be feasible, either in the primal or in the dual sense. On the other hand, it has the following property.

Lemma 3. $0 < \bar{a}_{k0} < 1$.

Proof. From (12),

$$\begin{aligned} &(u_{M_1}, -v_{M_2}) = (u_0 + v_0)e_k\hat{A}^{-1}, \\ &(u_0 + v_0)e_k\hat{A}^{-1}\hat{b} = v_0 \end{aligned} \tag{15}$$

and since $u_0 > 0$, $v_0 > 0$,

$$0 < \bar{a}_{k0} = e_k\hat{A}^{-1}\hat{b} = \frac{v_0}{u_0 + v_0} < 1. \qquad \square$$

Theorem 4A. *Let $\alpha x \geq \beta$ be the lift-and-project cut associated with a basic solution $(\alpha, \beta, u, u_0, v, v_0)$ to (CGLP)$_k$, with $u_0, v_0 > 0$, all components of α, β basic, and the basic components of u and v indexed by M_1 and M_2, respectively.*

Let $\pi s_J \geq \pi_0$ be the simple disjunctive cut from the disjunction $x_k \leq 0 \vee x_k \geq 1$ applied to (14) with $J := M_1 \cup M_2$.

Then $\pi s_J \geq \pi_0$ is equivalent to $\alpha x \geq \beta$.

Proof. From Lemma 2 we have that the matrix $\hat{A} = \tilde{A}_J$ is nonsingular, so (14) is well-defined. As stated in section 2, the cut $\pi s_J \geq \pi_0$ from the disjunction $x_k \leq 0 \vee x_k \geq 1$ applied to (14) is defined by

$$\pi_0 := \bar{a}_{k0}(1 - \bar{a}_{k0}) \tag{16}$$

where $\bar{a}_{k0} = e_k\hat{A}^{-1}\hat{b}$, and

$$\pi_j := \max\{\pi_j^1, \pi_j^2\}, \quad j \in J,$$

where

$$\pi_j^1 := \bar{a}_{kj}(1 - \bar{a}_{k0}) = -(\hat{A}^{-1})_{kj}(1 - \bar{a}_{k0}), \quad \pi_j^2 := -\bar{a}_{kj}\bar{a}_{k0} = (\hat{A}^{-1})_{kj}\bar{a}_{k0}. \tag{17}$$

The equivalence of $\pi s_J \geq \pi_0$ to the cut $\alpha x \geq \beta$ corresponding to the basic solution $w = (\alpha, \beta, u, u_0, v, v_0)$ of (CGLP)$_k$ is obtained by showing that

$$\begin{aligned} \theta\alpha &= \pi\hat{A}, & \theta\beta &= \pi_0 + \pi\hat{b}, \\ \theta u_J &= \pi - \pi^1, & \theta v_J &= \pi - \pi^2, \\ \theta u_0 &= 1 - \bar{a}_{k0}, & \theta v_0 &= \bar{a}_{k0}. \end{aligned} \tag{18}$$

for some $\theta > 0$. This we do by showing that $\alpha, \beta, u, u_0, v, v_0$ as defined by (18) satisfies (11). Indeed, using (16) and (17),

$$\begin{aligned} \theta(\alpha - u_J\hat{A} + u_0e_k) &= \pi\hat{A} - (\pi - \pi^1)\hat{A} + (1 - \bar{a}_{k0})e_k = \pi^1\hat{A} + (1 - \bar{a}_{k0})e_k \\ &= -e_k\hat{A}^{-1}\hat{A}(1 - \bar{a}_{k0}) + (1 - \bar{a}_{k0})e_k = 0 \\ \theta(-\beta + u_J\hat{b}) &= -(\pi_0 + \pi\hat{b}) + (\pi - \pi^1)\hat{b} = -\pi_0 - \pi^1\hat{b} \\ &= -\bar{a}_{k0}(1 - \bar{a}_{k0}) + e_k\hat{A}^{-1}\hat{b}(1 - \bar{a}_{k0}) \\ &= -\bar{a}_{k0}(1 - \bar{a}_{k0}) + \bar{a}_{k0}(1 - \bar{a}_{k0}) = 0 \\ \theta(\alpha - v_J\hat{A} - v_0e_k) &= \pi\hat{A} - (\pi - \pi^2)\hat{A} - \bar{a}_{k0}e_k = \pi^2\hat{A} - \bar{a}_{k0}e_k \\ &= e_k\hat{A}^{-1}\hat{A}\bar{a}_{k0} - \bar{a}_{k0}e_k = 0 \\ \theta(-\beta + v_J\hat{b} + v_0) &= -(\pi_0 + \pi\hat{b}) + (\pi - \pi^2)\hat{b} + \bar{a}_{k0} = -\pi_0 - \pi^2\hat{b} + \bar{a}_{k0} \\ &= -\bar{a}_{k0}(1 - \bar{a}_{k0}) - e_k\hat{A}^{-1}\hat{b}\bar{a}_{k0} + \bar{a}_{k0} \\ &= -\bar{a}_{k0}(1 - \bar{a}_{k0}) - \bar{a}_{k0}\bar{a}_{k0} + \bar{a}_{k0} = 0. \end{aligned}$$

We further have that

$$\theta u_j = \pi_j - \pi_j^1 = \max\{\pi_j^1, \pi_j^2\} - \pi_j^1 = \begin{cases} \pi_j^2 & \text{if } j \in M_1 \\ 0 & \text{if } j \in M_2 \end{cases}$$

and

$$\theta v_j = \pi_j - \pi_j^2 = \max\{\pi_j^1, \pi_j^2\} - \pi_j^2 = \begin{cases} 0 & \text{if } j \in M_1 \\ \pi_j^1 & \text{if } j \in M_2 \end{cases}$$

so u and v as defined by (18) is zero for every component not in M_1 and M_2 respectively. Finally, if we choose θ in (18) such that the normalization constraint

$$u_{M_1}\mathbf{1}_{M_1} + u_0 + v_{M_2}\mathbf{1}_{M_2} + v_0 = 1$$

is satisfied, then we have that $\alpha, \beta, u, u_0, v, v_0$ as defined by (18) satisfies the system (11). Since w is a basic solution and therefore is the unique solution to (11) then it must be as defined by (18).

The cut $\theta\alpha x \geq \theta\beta$ defined by (18) is $(\pi\hat{A})x \geq (\pi_0 + \pi\hat{b})$. Substituting for x using (13) we obtain the cut $\pi s_J \geq \pi_0$, which shows the equivalence. □

Theorem 4A has the following converse.

Theorem 4B. *Let $\hat{A}$ be any $n \times n$ nonsingular submatrix of $\tilde{A}$ and $\hat{b}$ the corresponding subvector of $\tilde{b}$, such that*

$$0 < e_k\hat{A}^{-1}\hat{b} < 1,$$

and let J be the row index set of $(\hat{A}, \hat{b})$. Further, let $\pi s_J \geq \pi_0$ be the simple disjunctive cut obtained from the disjunction $x_k \leq 0 \vee x_k \geq 1$ applied to the expression of x_k in terms of the nonbasic variables indexed by J. Further, let (M_1, M_2) be any partition of J such that $j \in M_1$ if $\pi_j^1 < \pi_j^2$ (i.e. $\bar{a}_{kj} < 0$) and $j \in M_2$ if $\pi_j^1 > \pi_j^2$ (i.e. $\bar{a}_{kj} > 0$), where π_j^1, π_j^2 are defined by (17).

Now let $\alpha x \geq \beta$ be the lift-and-project cut corresponding to the basic solution to (CGLP)$_k$ in which all components of α, β are basic, both u_0 and v_0 are positive, and the basic components of u and v are indexed by M_1 and M_2, respectively.

Then $\alpha x \geq \beta$ is equivalent to $\pi s_J \geq \pi_0$.

Proof. First we show that the choice of basic variables in the Theorem is well-defined, i.e., that they form a basis. We proceed as in the proof of Lemma 2 by first eliminating all variables chosen to be nonbasic, from the system (10), which results in the system (11). If we further eliminate α and β we obtain (12). Here $\binom{\tilde{A}_{M_1}}{\tilde{A}_{M_2}} = \hat{A}$, which is nonsingular. From this it follows that (12) has a unique solution and hence (11) also has a unique solution. Therefore, the choice of basic variables forms a basis.

Next we show the equivalence. Consider the solution $(\alpha, \beta, u, u_0, v, v_0)$ defined by (18). Following the proof of Theorem 4A we have that this solution satisfies (11) for a certain θ and that $\alpha x \geq \beta$ is equivalent to the cut $\pi s_J \geq \pi_0$. We have shown above that (11) has a unique solution for the choice of basis given in Theorem 4B, hence this basic solution must be as defined by (18). □

Note that in spite of the close correspondence that Theorems 4A and 4B establish between bases of $(\text{CGLP})_k$ and those of (LP), this correspondence is in general not one to one. Of course, when $\pi_j^1 \neq \pi_j^2$ for all $j \in J$, then the partition (M_1, M_2) of J is unique. But when $\pi_j^1 = \pi_j^2$ for some $j \in J$, which is only possible when $\pi_j^1 = \pi_j^2 = 0$ (since π_j^1 and π_j^2, when nonzero, are of opposite signs), then the corresponding index j can be assigned to either M_1 or M_2, each assignment yielding a different basis for $(\text{CGLP})_k$. However, although the two bases are different, the associated basic solutions to $(\text{CGLP})_k$ are the same, since the components of u and v corresponding to an index $j \in J$ with $\pi_j^1 = \pi_j^2 = 0$ are 0, and hence the pivot in $(\text{CGLP})_k$ that takes one basis into the other is degenerate; i.e., the change of bases does not produce a change of solutions.

The system (10) consists of a cone given by the homogeneous constraints truncated by a single hyperplane – the normalization constraint. Since (10) is bounded the extreme points of (10) are in one-to-one correspondence with the extreme rays of the non-normalized cone. Hence the relationship in Theorems 4A and 4B can also be interpreted as one between basic solutions to (LP) and extreme rays of the cone defined by the homogenous system of (10).

Next we show that the correspondence between the cuts $\alpha x \geq \beta$ and $\pi s_J \geq \pi_0$ established in Theorem 4A, 4B carries over to the strengthened version of these cuts.

5. The correspondence between the strengthened cuts

Theorem 5. *Theorems 4A and 4B remain valid if the inequalities $\alpha x \geq \beta$ and $\pi s_J \geq \pi_0$ are replaced by the strengthened lift-and-project cut $\bar{\alpha} x \geq \beta$ defined by (8), (9), and the mixed integer Gomory cut (or strengthened simple disjunctive cut) $\bar{\pi} s_J \geq \pi_0$ defined by (5), respectively.*

Proof. In the simplex tableau (2) corresponding to the nonbasic index set J we define $B := N \cap I$ and $R := N \cap J$ for the basic and nonbasic structural variables, respectively; as well as $P := S \cap I$ can $Q := S \cap J$ for the basic and nonbasic surplus variables, respectively.

The only coefficients of the cut $\pi s_J \geq \pi_0$ that can possibly be strengthened are those π_j such that $j \in J_1 := J \cap \{m + p + 1, \dots, m + p + n\}$, i.e. x_j is a structural integer-constrained variable, nonbasic in the simplex tableau (2). We claim that these are precisely the indices of the coefficients of the cut $\alpha x \geq \beta$ that can be strengthened. Indeed, by substituting for the surplus variables of the constraints $Ax \geq b$ that are tight, $\pi s_J \geq \pi_0$ can be written as

$$\pi_R s_R + \pi_Q (A_Q x - b_Q) \geq \pi_0$$

or

$$(\pi_R + \pi_Q A_Q^R) x_R + \pi_Q A_Q^B x_B \geq \pi_0 + \pi_Q b_Q.$$

Here $A_Q x \geq b_Q$ is the subsystem of $\hat{A} x \geq \hat{b}$ consisting of the rows indexed by Q. The components of π are $\pi_j := \max\{\pi_j^1, \pi_j^2\}$, where

$$\pi_j^1 := \bar{a}_{kj}(1 - \bar{a}_{k0}), \quad \pi_j^2 := -\bar{a}_{kj} \bar{a}_{k0},$$

with $\pi_0 := \bar{a}_{k0}(1 - \bar{a}_{k0})$.

On the other hand, writing $\alpha := (\alpha_B, \alpha_R)$, $A_Q := (A_Q^B, A_Q^R)$, from (18) we have

$$\begin{aligned} \alpha_B &= \pi_Q A_Q^B, \\ \alpha_R &= \pi_R + \pi_Q A_Q^R \\ \beta &= \pi_0 + \pi_Q b_Q. \end{aligned} \tag{19}$$

Here we assume that the solution $(\alpha, \beta, u, u_0, v, v_0)$ given by (18) is scaled such that $\theta = 1$ for ease of notation.

Also, $\alpha = \max\{\alpha^1, \alpha^2\}$, where max is the component-wise maximum, and,

$$\begin{aligned} \alpha^1 &= uA - u_0 e_k = \alpha - u_N, \\ \alpha^2 &= vA + v_0 e_k = \alpha - v_N, \end{aligned} \tag{20}$$

where u_N and v_N are the components of u and v, respectively, associated with the constraints $x_j \geq 0$, $j \in N$.

Now for $i \in B$, i.e. for the structural variables x_i basic in (2), the corresponding components of u and v are 0, i.e. $u_i = v_i = 0$, since by the choice of J it contains the index of every positive u_i and v_i. Thus we have that

$$\alpha_B^1 = \alpha_B^2 = \alpha_B,$$

and therefore the components of α corresponding to variables basic in (2) cannot be strengthened.

Consider now the components of α corresponding to the variables x_j nonbasic in (2). For $j \in R$ we have

$$\begin{aligned} \alpha_j^1 &= \alpha_j - u_j \quad \text{(from (20))} \\ &= \pi_j + (\pi_Q A_Q^R)_j - (\pi_j - \pi_j^1) \quad \text{(from (19) and (18))} \\ &= \pi_j^1 + \rho_j \end{aligned}$$

and

$$\begin{aligned} \alpha_j^2 &= \alpha_j - v_j \\ &= \pi_j^2 + \rho_j \end{aligned}$$

where ρ_j is the j-th column of $\pi_Q A_Q^R$.

Thus

$$\begin{aligned} \alpha_j &= \max\{\alpha_j^1, \alpha_j^2\} \\ &= \max\{\pi_j^1, \pi_j^2\} + \rho_j \end{aligned}$$

and the strengthened coefficient is

$$\bar{\alpha}_j = \min\{\alpha_j^1 + u_0 \lceil m_j \rceil,\ \alpha_j^2 - v_0 \lfloor m_j \rfloor\},$$

where

$$\begin{aligned} m_j &:= \tfrac{\alpha_j^2 - \alpha_j^1}{u_0 + v_0} \\ &= \pi_j^2 - \pi_j^1 \end{aligned}$$

since $u_0 + v_0 = 1 - \bar{a}_{k0} + \bar{a}_{k0} = 1$. Note that the definition of m_j does not depend on the scaling of $(\alpha, \beta, u, u_0, v, v_0)$. Consequently,

$$\begin{aligned}\bar{\alpha}_j &= \min\{\pi_j^1 + u_0\lceil m_j\rceil,\ \pi_j^2 - v_0\lfloor m_j\rfloor\} + \rho_j\\ &= \min\{\bar{a}_{kj}(1-\bar{a}_{k0}) + (1-\bar{a}_{k0})\lceil -\bar{a}_{kj}\rceil, -\bar{a}_{kj}\bar{a}_{k0} - \bar{a}_{k0}\lfloor -\bar{a}_{kj}\rfloor\} + \rho_j\\ &= \min\{f_{kj}(1-\bar{a}_{k0}), (1-f_{kj})\bar{a}_{k0}\} + \rho_j.\end{aligned}$$

Thus the coefficients $\bar{\alpha}_j$ are the same as the coefficients $\bar{\pi}_j$ expressed in terms of the structural variables. □

6. Bounds on the number of essential cuts

The correspondences established in Theorems 4A, 4B and 5 allow us to derive some new bounds on the number of undominated cuts of each type. Indeed, since every valid inequality for $\{x \in P : (x_k \le 0) \vee (x_k \ge 1)\}$ is dominated by some lift-and-project cut corresponding to a basic feasible solution of $(\text{CGLP})_k$, the number of undominated valid inequalities is bounded by the number of bases of $(\text{CGLP})_k$, which in turn cannot exceed

$$\binom{\#\text{ variables}}{\#\text{ constraints}} = \binom{2(m+p+n+1)+n+1}{2n+3},$$

a rather weak bound. But from Theorems 4A/4B it follows that a much tighter upper bound is also available, namely the number of ways to choose a subset of n variables to be nonbasic in a simplex tableau where x_k is basic, that is, the number of subsets J of cardinality n of the set $\{1, \ldots, m+p+n\} \setminus \{k\}$:

Corollary 6. *The number of facets of the polyhedron*

$$P_k := \text{conv}\,\{x \in P : x \textit{ satisfies } (6)_k\}$$

is bounded by

$$\binom{m+p+n-1}{n}. \qquad \square$$

Thus the elementary closure $\bigcap_{k=1}^{p} P_k$ of P with respect to the lift-and-project operation has at most $p\binom{m+p+n-1}{n}$ facets.

Similarly, the number of undominated simple disjunctive cuts obtainable by applying the disjunction $x_k \le 0 \vee x_k \ge 1$ to the expression of x_k in terms of any other variables is bounded by

$$\binom{m+p+n-1}{n},$$

and the number of cuts of this type for all $k \in \{1, \ldots, p\}$ is consequently at most $p\binom{m+p+n-1}{n}$.

If we try to extend these bounds to strengthened lift-and-project cuts, or to strengthened simple disjunctive cuts, we run into the problem that the extension is only valid if we restrict ourselves to strengthened cuts derived from basic solutions. But this is not satisfactory, since although any unstrengthened cut in either class is dominated by some unstrengthened cut corresponding to a basic solution, the same is not true of the strengthened cuts: a strengthened cut from a nonbasic solution need not be dominated by any strengthened cut from a basic solution: counterexamples are easy to produce.

On the other hand, the correspondences established in Theorems 4A/4B and 5 have an important consequence on the rank of the LP relaxation of a 0-1 mixed integer program with respect to the various families of cuts examined here.

7. The rank of *P* with respect to different cuts

Let P again denote the feasible region of the LP relaxation, as defined in section 1. It is well known that in the case of a pure 0-1 program, i.e. when $p = n$, the rank of P with respect to the family of (pure integer) fractional Gomory cuts can be strictly greater than n [11]. We now show that by contrast, the rank of P with respect to the family of mixed integer Gomory cuts is at most p.

We first recall the definition of rank. We say that P has rank k with respect to a certain family of cuts (or with respect to a certain cut generating procedure) if k is the smallest integer such that, starting with P and applying the cut generating procedure recursively k times, yields the convex hull of 0-1 points in P.

Theorem 7. *The rank of P with respect to each of the following families of cuts is at most p, the number of 0-1 variables:*

(a) unstrengthened lift-and-project cuts;
(b) simple disjunctive cuts;
(c) strengthened lift-and-project cuts;
(d) mixed integer Gomory cuts or, equivalently, strengthened simple disjunctive cuts.

Proof. Denote, as before, $P := \{x \in \mathbb{R}^n : \tilde{A}x \geq \tilde{b}\}$, and

$$P_D := \text{conv}\,\{x \in P : x_j \in \{0, 1\}, j = 1, \ldots, p\}.$$

From the basic result of disjunctive programming on sequential convexification [2, 3], if we define $P_0 := P$ and for $j = 1, \ldots, p$,

$$P^j := \text{conv}\,\{P^{j-1} \cap \{x \in \mathbb{R}^n : x_j \in \{0, 1\}\}),$$

then

$$P^p = P_D.$$

Since P^j can be obtained from P^{j-1} by unstrengthened lift-and-project cuts, this implies (a).

From Theorems 4A/4B, at any iteration j of the above procedure, each lift-and-project cut used to generate P^j, corresponding to some basic solution of $(\text{CGLP})_j$, can also be obtained as a simple disjunctive cut associated with some nonbasic set J, with $|J| = n$. Hence the whole sequential convexification procedure can be stated in terms of simple disjunctive cuts rather than lift-and-project cuts, which implies (b).

Turning now to strengthened lift-and-project cuts, if at each iteration j of the above procedure we use strengthened rather than unstrengthened lift-and-project cuts corresponding to basic solutions of $(\text{CGLP})_j$, we obtain a set $\tilde{P}^j$ instead of P^j, with $\tilde{P}^j \subseteq P^j$. Clearly, using the same recursion as above, we end up with $\tilde{P}^p = P_D$. This proves (c).

Finally, since every strengthened lift-and-project cut corresponding to a basic solution of $(\text{CGLP})_j$ is equivalent to a mixed integer Gomory cut derived from the row corresponding to x_j of a simplex tableau with a certain nonbasic set J with $|J| = n$ (Theorem 5), the procedure discussed under (c) can be restated as an equivalent procedure in terms of mixed integer Gomory cuts, which proves (d). □

In [9] it was shown that the bound established in Theorem 7 is tight for the mixed integer Gomory cuts, by providing a class of examples with rank p.

We now turn to the computational implications of Theorems 4A/4B and 5.

8. Solving $(\text{CGLP})_k$ on the (LP) simplex tableau

The major practical consequence of the correspondence established in Theorems 4A/4B is that the cut generating linear program $(\text{CGLP})_k$ need not be formulated and solved explicitly; instead, the procedure for solving it can be mimicked on the linear programming relaxation (LP) of the original mixed 0-1 problem. Apart from the fact that this replaces a large linear program with a smaller one, it also substantially reduces the number of pivots for the following reason. A basic solution to (LP) associated with a nonbasic set J corresponds to a *set* of basic solutions to $(\text{CGLP})_k$ having $u_0 > 0$, $v_0 > 0$, all components of (α, β) basic, and u_i, v_j basic for some $i \in M_1$, and $j \in M_2$, respectively, such that $M_1 \cup M_2 = J$. The various solutions to $(\text{CGLP})_k$ that correspond to the basic solution to (LP) associated with J differ among themselves by the partition of J into M_1 and M_2. These solutions can be obtained from each other by degenerate pivots in $(\text{CGLP})_k$. Thus a single pivot in (LP), which replaces the set J with some J' that differs from J in a single element, may correspond to several pivots in $(\text{CGLP})_k$, which together change the set $M_1 \cup M_2$ by a single element, but shift one or more elements from M_1 to M_2 and vice-versa.

We will now describe the procedure that mimics on (LP) the optimization of $(\text{CGLP})_k$. We start with the simple disjunctive cut $\pi s_J \geq \pi_0$ derived from the optimal simplex tableau (2) by applying the disjunction $x_k \leq 0 \vee x_k \geq 1$ to the expression

$$x_k = \bar{a}_{k0} - \sum_{j \in J} \bar{a}_{kj} s_j. \tag{21}$$

(same as (14)). As mentioned before, the coefficients of this cut are

$$\pi_0 := (1 - \bar{a}_{k0})\bar{a}_{k0}$$

and

$$\pi_j := \max\{\pi_j^1, \pi_j^2\}, \qquad j \in J,$$

with $\pi_j^1 := (1 - \bar{a}_{k0})\bar{a}_{kj}$, $\pi_j^2 := -\bar{a}_{k0}\bar{a}_{kj}$.

We know that the lift-and-project cut $\alpha x \geq \beta$ equivalent to $\pi s_j \geq \pi_0$ corresponds to the basic solution of $(\text{CGLP})_k$ defined by (18). We wish to obtain the lift-and-project cut corresponding to an optimal solution to $(\mathrm{CGLP})_k$ by performing the improving pivots in (LP).

We start by examining a pivot on an element $\bar{a}_{ij}$, $i \neq k$ of the simplex tableau (2) of (LP). The effect of such a pivot is to add to the cut row (21) the i-th row multiplied by $\gamma_j := -\bar{a}_{kj}/\bar{a}_{ij}$, and hence to replace (21) by

$$x_k = \bar{a}_{k0} + \gamma_j \bar{a}_{i0} - \sum_{h \in J \setminus \{j\}} (\bar{a}_{kh} + \gamma_j \bar{a}_{ih}) s_h - \gamma_j x_i \tag{22}$$

Now if $0 < \bar{a}_{k0} + \gamma_j \bar{a}_{i0} < 1$, i.e. if $(-\bar{a}_{k0}/\bar{a}_{i0}) < \gamma_j < ((1 - \bar{a}_{k0})/\bar{a}_{i0})$, then we can apply the disjunction $x_k \leq 0 \vee x_k \geq 1$ to (22) instead of (21), to obtain a cut $\pi^\gamma s_{J^\gamma} \geq \pi_0^\gamma$, where $J^\gamma := (J \setminus \{j\}) \cup \{i\}$ and s_i denotes x_i. The question is how to choose the pivot element $\bar{a}_{ij}$ in order to make $\pi^\gamma s_{J^\gamma} \geq \pi_0^\gamma$ a stronger cut than $\pi s_J \geq \pi_0$, in fact as strong as possible. This choice involves two elements. First, we choose a row i, some multiple of which is to be added to row k; second, we choose a column in row i, which sets the sign and size of the multiplier. Note that we can pivot on *any* nonzero $\bar{a}_{ij}$ since we do not restrict ourselves to feasible bases.

As to the first choice, pivoting in row i, i.e. pivoting the variable x_i out of the basis, corresponds in the simplex tableau for $(\mathrm{CGLP})_k$ to pivoting into the basis one of the variables u_i or v_i. Clearly, such a pivot is an improving one in terms of the objective function of $(\mathrm{CGLP})_k$ only if either u_i or v_i have a negative reduced cost. Below we give the expressions for r_{u_i} and r_{v_i}, the reduced costs of u_i and v_i respectively, in terms of the coefficients $\bar{a}_{kj}$ and $\bar{a}_{ij}$, $j \in J \cup \{0\}$.

As to the second choice, one can identify the index $j \in J$ such that pivoting on $\bar{a}_{ij}$ maximizes the improvement in the strength of the cut, by first maximizing the improvement over all $j \in J$ with $\gamma_j = -\bar{a}_{ij}/\bar{a}_{ij} > 0$, then over all $j \in J$ with $\gamma_j < 0$ and choosing the larger of the two improvements.

In the following we will consider a basic solution (x, s) to (LP). In the simplex tableau (2) corresponding to this solution we let, as earlier, $B := N \cap I$ and $R := N \cap J$ denote the basic and nonbasic structural variables, respectively, and we let $P := S \cap I$ and $Q := S \cap J$ denote the basic and nonbasic surplus variables, respectively.

Lemma 8. *In the simplex tableau (2) corresponding to the basic solution (x, s), the coefficients $\bar{a}_{ij}$ for $i = 1, \ldots, m + p + n$, $j \in J$, and the right-hand sides $\bar{a}_{i0}$ for $i = 1, \ldots, m + p + n$ satisfy*

$$\bar{a}_{ij} = -(\tilde{A}_i \tilde{A}_J^{-1})_j \tag{23}$$

$$\bar{a}_{i0} = \tilde{A}_i \tilde{A}_J^{-1} \tilde{b}_J - \tilde{b}_i \tag{24}$$

Proof. From Lemma 2 we know that $\tilde{A}_J$ is invertible. Let us first consider the basis matrix for (LP) and its inverse. We can write the nontrivial constraints of (LP) as

$$\begin{aligned} A_Q^B x_B \qquad\quad &+ A_Q^R x_R - s_Q = b_Q \\ A_P^B x_B - s_P &+ A_P^R x_R \qquad\;\; = b_P \end{aligned}$$

where $A = (A^B \; A^R)$. The basis matrix, E, for (LP) and its inverse, E^{-1}, are

$$E = \begin{pmatrix} A_Q^B & 0 \\ A_P^B & -I \end{pmatrix} \qquad E^{-1} = \begin{pmatrix} (A_Q^B)^{-1} & 0 \\ A_P^B (A_Q^B)^{-1} & -I \end{pmatrix}$$

The constraints of $\tilde{A}x \geq \tilde{b}$ indexed by J are

$$\begin{aligned} A_Q^B x_B + A_Q^R x_R - s_Q \qquad &= b_Q \\ x_R \qquad - s_R &= 0 \end{aligned}$$

where s_R are those surplus variables of (1) corresponding to x_R.

We can thus write $\tilde{A}_J$ and its inverse, $\tilde{A}_J^{-1}$ as

$$\tilde{A}_J = \begin{pmatrix} A_Q^B & A_Q^R \\ 0 & I \end{pmatrix} \qquad \tilde{A}_J^{-1} = \begin{pmatrix} (A_Q^B)^{-1} & -(A_Q^B)^{-1} A_Q^R \\ 0 & I \end{pmatrix}$$

There are four cases to consider for the coefficients of the simplex tableau for (LP); index i can be either that of a structural variable ($i \in B$) or that of a surplus variable ($i \in P$), and likewise for index j ($j \in R$ or $j \in Q$).

Case 1. i and j both index a surplus variable ($i \in P,\ j \in Q$). We obtain $\bar{a}_{ij}$ by pre-multiplying the j-th nonbasic column, which in this case is the column for surplus variable j, by E^{-1} and taking the i-th component, as

$$\bar{a}_{ij} = (0\ e_i)E^{-1}\begin{pmatrix} -e_j \\ 0 \end{pmatrix} = -(A_P^B (A_Q^B)^{-1})_{ij} = -(A_i^B (A_Q^B)^{-1})_j,$$

where A_i^B is the i-th row of A^B. To show (23), note that for this case we have $\tilde{A}_i = (A_i^B\ A_i^R)$ and the j-th component of $-\tilde{A}_i \tilde{A}_J^{-1}$ becomes

$$-\tilde{A}_i \tilde{A}_J^{-1}\begin{pmatrix} e_j \\ 0 \end{pmatrix} = -(A_i^B\ A_i^R)\tilde{A}_J^{-1}\begin{pmatrix} e_j \\ 0 \end{pmatrix} = -A_i^B (A_Q^B)^{-1} e_j = -(A_i^B (A_Q^B)^{-1})_j = \bar{a}_{ij}.$$

Case 2. i indexes a surplus variable and j indexes a structural variable ($i \in P,\ j \in R$). Again $\tilde{A}_i = (A_i^B\ A_i^R)$, and

$$\begin{aligned} \bar{a}_{ij} &= (0\ e_i)E^{-1}\begin{pmatrix} A_Q^R \\ A_P^R \end{pmatrix}_j = A_i^B (A_Q^B)^{-1}(A_Q^R)_j - A_{ij}^R \\ -\tilde{A}_i \tilde{A}_J^{-1}\begin{pmatrix} 0 \\ e_j \end{pmatrix} &= -(A_i^B\ A_i^R)\tilde{A}_J^{-1}\begin{pmatrix} 0 \\ e_j \end{pmatrix} = A_i^B (A_Q^B)^{-1}(A_Q^R)_j - A_{ij}^R = \bar{a}_{ij}. \end{aligned}$$

Case 3. i indexes a structural variable and j indexes a surplus variable ($i \in B,\ j \in Q$). Here $\tilde{A}_i = (e_i\ 0)$, and

$$\begin{aligned} \bar{a}_{ij} &= (e_i\ 0)E^{-1}\begin{pmatrix} -e_j \\ 0 \end{pmatrix} = -((A_Q^B)^{-1})_{ij} \\ -\tilde{A}_i \tilde{A}_J^{-1}\begin{pmatrix} e_j \\ 0 \end{pmatrix} &= -(e_i\ 0)\tilde{A}_J^{-1}\begin{pmatrix} e_j \\ 0 \end{pmatrix} = -((A_Q^B)^{-1})_{ij} = \bar{a}_{ij}. \end{aligned}$$

Case 4. i and j both index a structural variable ($i \in B,\ j \in R$). Again $\tilde{A}_i = (e_i\ 0)$, and

$$\bar{a}_{ij} = (e_i\ 0)E^{-1}\begin{pmatrix} A_Q^R \\ A_P^R \end{pmatrix}_j = ((A_Q^B)^{-1}A_Q^R)_{ij}$$

$$-\tilde{A}_i\tilde{A}_J^{-1}\begin{pmatrix} 0 \\ e_j \end{pmatrix} = -(e_i\ 0)\tilde{A}_J^{-1}\begin{pmatrix} 0 \\ e_j \end{pmatrix} = ((A_Q^B)^{-1}A_Q^R)_{ij} = \bar{a}_{ij}.$$

For the right-hand side $\bar{a}_{i0}$ there are only two cases, depending on whether i indexes a structural or a surplus variable.

Case 1. i indexes a surplus ($i \in P$). Then

$$\bar{a}_{i0} = (0\ e_i)E^{-1}\begin{pmatrix} b_Q \\ b_P \end{pmatrix} = A_i^B(A_Q^B)^{-1}b_Q - b_i$$

To show (24), note that $\tilde{b}_i = b_i$ and

$$\tilde{A}_i\tilde{A}_J^{-1}\begin{pmatrix} \tilde{b}_Q \\ \tilde{b}_R \end{pmatrix} - \tilde{b}_i = (A_i^B\ A_i^R)\tilde{A}_J^{-1}\begin{pmatrix} b_Q \\ 0 \end{pmatrix} - b_i = A_i^B(A_Q^B)^{-1}b_Q - b_i = \bar{a}_{i0}$$

Case 2. i indexes a structural variable ($i \in B$). Then

$$\bar{a}_{i0} = (e_i\ 0)E^{-1}\begin{pmatrix} b_Q \\ b_P \end{pmatrix} = ((A_Q^B)^{-1}b_Q)_i,$$

and since $\tilde{b}_i = 0$,

$$\tilde{A}_i\tilde{A}_J^{-1}\begin{pmatrix} \tilde{b}_Q \\ \tilde{b}_R \end{pmatrix} - \tilde{b}_i = (e_i\ 0)\tilde{A}_J^{-1}\begin{pmatrix} b_Q \\ 0 \end{pmatrix} - 0 = ((A_Q^B)^{-1}b_Q)_i = \bar{a}_{i0}. \qquad \square$$

Recall that $\bar{a}_{ij}$ and $\bar{a}_{i0}$ denotes respectively the coefficient for variable j in row i and the right-hand side of row i, in the simplex tableau of (LP) for the *current* solution (x, s), whereas $(\bar{x}, \bar{s})$ is the *optimal* solution to (LP).

Theorem 9. *Let $(\alpha, \beta, u, u_0, v, v_0)$ be a basic feasible solution to (10) with $u_0, v_0 > 0$, all components of α, β basic, and the basic components of u and v indexed by M_1 and M_2, respectively. Let $\bar{s}$ be the surplus variables of $\tilde{A}x \geq \tilde{b}$ corresponding to the solution $\bar{x}$.*

The reduced costs of u_i and v_i for $i \notin J \cup \{k\}$ in this basic solution are, respectively

$$\begin{aligned} r_{u_i} &= \sigma\left(-\textstyle\sum_{j\in M_1}\bar{a}_{ij} + \sum_{j\in M_2}\bar{a}_{ij} - 1\right) - \sum_{j\in M_2}\bar{a}_{ij}\bar{s}_j + \bar{a}_{i0}(1-\bar{x}_k) \\ r_{v_i} &= \sigma\left(+\textstyle\sum_{j\in M_1}\bar{a}_{ij} - \sum_{j\in M_2}\bar{a}_{ij} - 1\right) - \sum_{j\in M_1}\bar{a}_{ij}\bar{s}_j + \bar{a}_{i0}\bar{x}_k \end{aligned} \tag{25}$$

where

$$\sigma = \frac{\sum_{j\in M_2}\bar{a}_{kj}\bar{s}_j - \bar{a}_{k0}(1-\bar{x}_k)}{1 + \sum_{j\in J}|\bar{a}_{kj}|}$$

Proof. If we restrict (10) to the basic variables plus u_i and v_i, and eliminate α, β, we obtain the system

$$\begin{aligned}
&u_{M_1}\tilde{A}_{M_1} + u_i\tilde{A}_i - u_0 e_k = v_{M_2}\tilde{A}_{M_2} + v_i\tilde{A}_i + v_0 e_k\\
&u_{M_1}\tilde{b}_{M_1} + u_i\tilde{b}_i = v_{M_2}\tilde{b}_{M_2} + v_i\tilde{b}_i + v_0\\
&\textstyle\sum_{j\in M_1} u_j + u_i + u_0 + \sum_{j\in M_2} v_j + v_i + v_0 = 1
\end{aligned}$$

The first two equations can be rewritten

$$\begin{aligned}
(u_{M_1}, -v_{M_2})\tilde{A}_J + (u_i - v_i)\tilde{A}_i &= (u_0 + v_0)e_k\\
(u_{M_1}, -v_{M_2})\tilde{b}_J + (u_i - v_i)\tilde{b}_i &= v_0
\end{aligned}$$

From Lemma 2 we know that $\tilde{A}_J$ is invertible, so

$$\begin{aligned}
(u_{M_1}, -v_{M_2}) &= (u_0 + v_0)e_k\tilde{A}_J^{-1} - (u_i - v_i)\tilde{A}_i\tilde{A}_J^{-1}\\
v_0 &= (u_0 + v_0)e_k\tilde{A}_J^{-1}\tilde{b}_J - (u_i - v_i)(\tilde{A}_i\tilde{A}_J^{-1}\tilde{b}_J - \tilde{b}_i)
\end{aligned} \tag{26}$$

Now, using Lemma 8 we can identify the expressions $-(e_k\tilde{A}_J^{-1})_j$ and $-(\tilde{A}_i\tilde{A}_J^{-1})_j$ with the coefficients $\bar{a}_{kj}$ (since $\tilde{A}_k = e_k$) and $\bar{a}_{ij}$ in the simplex tableau of (LP) for the basic solution with variables indexed by J being nonbasic. Likewise we can identify the expressions $e_k\tilde{A}_J^{-1}\tilde{b}_J$ and $\tilde{A}_i\tilde{A}_J^{-1}\tilde{b}_J - \tilde{b}_i$ with the right-hand side constants $\bar{a}_{k0}$ (since $\tilde{b}_k = 0$) and $\bar{a}_{i0}$ of the simplex tableau. With this substitution we have that

$$\begin{aligned}
u_j &= -(u_0 + v_0)\bar{a}_{kj} + (u_i - v_i)\bar{a}_{ij} \quad \text{for } j \in M_1\\
v_j &= (u_0 + v_0)\bar{a}_{kj} - (u_i - v_i)\bar{a}_{ij} \quad \text{for } j \in M_2\\
v_0 &= (u_0 + v_0)\bar{a}_{k0} - (u_i - v_i)\bar{a}_{i0}
\end{aligned} \tag{27}$$

We can now write the normalization constraint as

$$1 = \textstyle\sum_{j\in M_1} u_j + \sum_{j\in M_2} v_j + u_i + v_i + u_0 + v_0$$

(substituting u_j and v_j from (27))

$$\begin{aligned}
&= \textstyle\sum_{j\in M_1}\big(-(u_0 + v_0)\bar{a}_{kj} + (u_i - v_i)\bar{a}_{ij}\big)\\
&\quad + \textstyle\sum_{j\in M_2}\big((u_0 + v_0)\bar{a}_{kj} - (u_i - v_i)\bar{a}_{ij}\big)\\
&\quad + u_i + v_i + u_0 + v_0\\
&= (u_0 + v_0)\Big(-\textstyle\sum_{j\in M_1}\bar{a}_{kj} + \sum_{j\in M_2}\bar{a}_{kj} + 1\Big)\\
&\quad + (u_i - v_i)\Big(\textstyle\sum_{j\in M_1}\bar{a}_{ij} - \sum_{j\in M_2}\bar{a}_{ij}\Big) + u_i + v_i.
\end{aligned}$$

Since (27) is satisfied for the current basic solution with $u_i = v_i = 0$ and since $u_{M_1}, v_{M_2} \geq 0$, it follows from (27) that $j \in M_1 \Rightarrow \bar{a}_{kj} \leq 0$ and $j \in M_2 \Rightarrow \bar{a}_{kj} \geq 0$, so

$$-\sum_{j\in M_1}\bar{a}_{kj} + \sum_{j\in M_2}\bar{a}_{kj} = \sum_{j\in J}|\bar{a}_{kj}|$$

We thus have

$$u_0 + v_0 = \frac{1 - (u_i - v_i)\Big(\sum_{j\in M_1}\bar{a}_{ij} - \sum_{j\in M_2}\bar{a}_{ij}\Big) - u_i - v_i}{1 + \sum_{j\in J}|\bar{a}_{kj}|} \tag{28}$$

We can now write the objective function of $(\text{CGLP})_k$ in terms of u_i and v_i as

$$\begin{aligned}
\alpha\bar{x} - \beta &= v_{M_2}(\tilde{A}_{M_2}\bar{x} - \tilde{b}_{M_2}) + v_i(\tilde{A}_i\bar{x} - \tilde{b}_i) \\
&\quad + v_0(e_k\bar{x} - 1)
\end{aligned}$$

(use that $\bar{s}_{M_2} = \tilde{A}_{M_2}\bar{x} - \tilde{b}_{M_2}$ and $\bar{s}_i = \tilde{A}_i\bar{x} - \tilde{b}_i$)

$$= v_{M_2}\bar{s}_{M_2} + v_i\bar{s}_i + v_0(\bar{x}_k - 1)$$

(substitute for v_{M_2} and v_0 using (27))

$$\begin{aligned}
&= (u_0 + v_0)\textstyle\sum_{j\in M_2}\bar{a}_{kj}\bar{s}_j \\
&\quad -(u_i - v_i)\textstyle\sum_{j\in M_2}\bar{a}_{ij}\bar{s}_j + v_i\bar{s}_i \\
&\quad +(u_0+v_0)\bar{a}_{k0}(\bar{x}_k - 1) - (u_i - v_i)\bar{a}_{i0}(\bar{x}_k - 1) \\
&= (u_0 + v_0)\left(\textstyle\sum_{j\in M_2}\bar{a}_{kj}\bar{s}_j - \bar{a}_{k0}(1 - \bar{x}_k)\right) \\
&\quad +(u_i - v_i)\left(-\textstyle\sum_{j\in M_2}\bar{a}_{ij}\bar{s}_j + \bar{a}_{i0}(1 - \bar{x}_k)\right) \\
&\quad +v_i\bar{s}_i
\end{aligned}$$

If we substitute for $(u_0 + v_0)$ from (28) and use the definition of σ we obtain

$$\begin{aligned}
\alpha\bar{x} - \beta \ &= \sigma + u_i\left(-\sigma\textstyle\sum_{j\in M_1}\bar{a}_{ij} + \sigma\sum_{j\in M_2}\bar{a}_{ij} - \sigma\right. \\
&\quad \left. - \textstyle\sum_{j\in M_2}\bar{a}_{ij}\bar{s}_j + \bar{a}_{i0}(1 - \bar{x}_k)\right) \\
&\quad +v_i\left(+\sigma\textstyle\sum_{j\in M_1}\bar{a}_{ij} - \sigma\sum_{j\in M_2}\bar{a}_{ij} - \sigma\right. \\
&\quad \left. + \textstyle\sum_{j\in M_2}\bar{a}_{ij}\bar{s}_j - \bar{a}_{i0}(1 - \bar{x}_k) + \bar{s}_i\right).
\end{aligned}$$

Substituting for $\bar{s}_i$ from $\bar{s}_i = \bar{a}_{i0} - \sum_{j\in M_1}\bar{a}_{ij}\bar{s}_j - \sum_{j\in M_2}\bar{a}_{ij}\bar{s}_j$, replaces the expression in the parentheses following v_i by $\sigma\sum_{j\in M_1}\bar{a}_{ij} - \sigma\sum_{j\in M_2}\bar{a}_{ij} - \sigma - \sum_{j\in M_1}\bar{a}_{ij}\bar{s}_j + \bar{a}_{i0}\bar{x}_k$. We can then read the reduced costs r_{u_i} and r_{v_i} as the coefficients of u_i and v_i. □

Theorem 10. *The pivot column in row i of the (LP) simplex tableau that is most improving with respect to the cut from row k, is indexed by that $l^* \in J$ that minimizes $f^+(\gamma_l)$ if $\bar{a}_{kl}\bar{a}_{il} < 0$ or $f^-(\gamma_l)$ if $\bar{a}_{kl}\bar{a}_{il} > 0$, over all $l \in J$ that satisfy $\frac{-\bar{a}_{k0}}{\bar{a}_{i0}} < \gamma_l < \frac{1-\bar{a}_{k0}}{\bar{a}_{i0})}$, where $\gamma_l := -\frac{\bar{a}_{kl}}{\bar{a}_{il}}$, and for $0 \le \gamma < \frac{1-\bar{a}_{k0}}{\bar{a}_{i0}}$*

$$f^+(\gamma) := \frac{\sum_{j\in J}(-(\bar{a}_{k0} + \gamma\bar{a}_{i0})\bar{a}_{kj} + \max\{\bar{a}_{kj}, -\gamma\bar{a}_{ij}\})\bar{x}_j - (1 - \bar{a}_{k0} - \gamma\bar{a}_{i0})\bar{a}_{k0}}{1 + |\gamma| + \sum_{j\in J}|\bar{a}_{kj} + \gamma\bar{a}_{ij}|},$$

and for $\frac{-\bar{a}_{k0}}{\bar{a}_{i0}} < \gamma \le 0$

$$f^-(\gamma) := \frac{\sum_{j\in J}(-(\bar{a}_{k0} + \gamma\bar{a}_{i0})\bar{a}_{kj} + \max\{\bar{a}_{kj} + \gamma\bar{a}_{ij}, 0\})\bar{x}_j - (1 - \bar{a}_{k0})(\bar{a}_{k0} + \gamma\bar{a}_{i0})}{1 + |\gamma| + \sum_{j\in J}|\bar{a}_{kj} + \gamma\bar{a}_{ij}|}$$

Proof. Consider row k and row i of the (LP) simplex tableau

$$x_k + \sum_{j\in J}\bar{a}_{kj}s_j = \bar{a}_{k0} \qquad x_i + \sum_{j\in J}\bar{a}_{ij}s_j = \bar{a}_{i0}.$$

If we add row i to row k with weight $\gamma \in \mathbb{R}$ we obtain the composite row

$$x_k + \gamma x_i + \sum_{j \in J} (\bar{a}_{kj} + \gamma \bar{a}_{ij}) s_j = \bar{a}_{k0} + \gamma \bar{a}_{i0}. \tag{29}$$

From (29) we can derive a simple disjunctive cut $\pi^\gamma s_J \geq \pi_0^\gamma$ using the disjunction $x_k \leq 0 \vee x_k \geq 1$ if the right-hand side of (29) satisfies $0 < \bar{a}_{k0} + \gamma \bar{a}_{i0} < 1$, i.e., if γ satisfies $\frac{-\bar{a}_{k0}}{\bar{a}_{i0}} < \gamma < \frac{1-\bar{a}_{k0}}{\bar{a}_{i0}}$.

A pivot on column l in row i has the effect of adding $\gamma_l = -\frac{\bar{a}_{kl}}{\bar{a}_{il}}$ times row i to row k. For this value of γ, (29) becomes the k-th row of the simplex tableau resulting from the pivot. We want to identify the column l such that the simple disjunctive cut, $\pi^\gamma s_J \geq \pi_0^\gamma$ we derive from the composite row (29) with $\gamma = \gamma_l$ minimizes $\pi^\gamma \bar{s}_J - \pi_0^\gamma$.

For any γ in the interval $\frac{-\bar{a}_{k0}}{\bar{a}_{i0}} < \gamma < \frac{1-\bar{a}_{k0}}{\bar{a}_{i0}}$, the simple disjunctive cut from the disjunction $x_k \leq 0 \vee x_k \geq 0$ applied to (29) has coefficients (see Section 2)

$$\begin{aligned}
\pi_i^\gamma &= \max\{(1 - \bar{a}_{k0} - \gamma \bar{a}_{i0})\gamma, -(\bar{a}_{k0} + \gamma \bar{a}_{i0})\gamma\} \\
\pi_j^\gamma &= \max\{(1 - \bar{a}_{k0} - \gamma \bar{a}_{i0})(\bar{a}_{kj} + \gamma \bar{a}_{ij}), -(\bar{a}_{k0} + \gamma \bar{a}_{i0})(\bar{a}_{kj} + \gamma \bar{a}_{ij})\} \text{ for } j \in J \\
\pi_0^\gamma &= (1 - \bar{a}_{k0} - \gamma \bar{a}_{i0})(\bar{a}_{k0} + \gamma \bar{a}_{i0})
\end{aligned}$$

Given the simple disjunctive cut $\pi_i^\gamma x_i + \pi^\gamma s_J \geq \pi_0^\gamma$, there exists a corresponding solution $(\alpha, \beta, u, u_0, v, v_0)$ to (CGLP)$_k$ given by (18). The corresponding π^1 and π^2 in (17) are

$$\begin{aligned}
\pi_i^{\gamma,1} &= (1 - \bar{a}_{k0} - \gamma \bar{a}_{i0})\gamma & \pi_j^{\gamma,1} &= (1 - \bar{a}_{k0} - \gamma \bar{a}_{i0})(\bar{a}_{kj} + \gamma \bar{a}_{ij}) & \\
\pi_i^{\gamma,2} &= -(\bar{a}_{k0} + \gamma \bar{a}_{i0})\gamma & \pi_j^{\gamma,2} &= -(\bar{a}_{k0} + \gamma \bar{a}_{i0})(\bar{a}_{kj} + \gamma \bar{a}_{ij}) & \text{for } j \in J
\end{aligned}$$

Then using (18) we have that the components (u, u_0, v, v_0) of this solution satisfy

$$\begin{aligned}
u_i + v_i &= (\pi_i^\gamma - \pi_i^{\gamma,1}) + (\pi_i^\gamma - \pi_i^{\gamma,2}) \\
&= |\pi_i^{\gamma,1} - \pi_i^{\gamma,2}| = |\gamma| \\
u_j + v_j &= (\pi_j^\gamma - \pi_j^{\gamma,1}) + (\pi_j^\gamma - \pi_j^{\gamma,2}) \\
&= |\pi_j^{\gamma,1} - \pi_j^{\gamma,2}| = |\bar{a}_{kj} + \gamma \bar{a}_{ij}| \quad \text{for } j \in J \\
u_0 + v_0 &= (1 - \bar{a}_{k0} - \gamma \bar{a}_{i0}) + (\bar{a}_{k0} + \gamma \bar{a}_{i0}) = 1
\end{aligned} \tag{30}$$

The solution to (CGLP)$_k$ corresponding to the cut $\pi_i^\gamma x_i + \pi^\gamma s_J \geq \pi_0^\gamma$ satisfies all the constraints of (CGLP)$_k$ except the normalization constraint. To also satisfy the normalization constraint we have to scale the cut by the sum of the multipliers u, v, u_0 and v_0, which from (30) becomes

$$\sum_{i=1}^{m+p+n} u_i + u_0 + \sum_{i=1}^{m+p+n} v_i + v_0 = 1 + |\gamma| + \sum_{j \in J} |\bar{a}_{kj} + \gamma \bar{a}_{ij}|$$

The objective function of (CGLP)$_k$ corresponding to this cut is thus obtained by scaling the function $\pi_i^\gamma \bar{x}_i + \pi^\gamma \bar{s}_J - \pi_0^\gamma$ by the sum of multipliers, namely

$$\frac{\pi_i^\gamma \bar{x}_i + \pi^\gamma \bar{s}_J - \pi_0^\gamma}{1 + |\gamma| + \sum_{j \in J} |\bar{a}_{kj} + \gamma \bar{a}_{ij}|} \tag{31}$$

The expressions for π^γ and π_0^γ involve terms with γ^2. We can eliminate such terms by subtracting π_i^γ times row i from the cut $\pi_i^\gamma x_i + \pi^\gamma s_J \geq \pi_0^\gamma$. The result depends on the sign of γ, since $\pi_i^\gamma = (1 - \bar{a}_{k0} - \gamma \bar{a}_{i0})\gamma$ if $\gamma > 0$, and $\pi_i^\gamma = -(\bar{a}_{k0} + \gamma \bar{a}_{i0})\gamma$ if $\gamma < 0$. Thus,

$$\gamma > 0 : \quad \begin{aligned} \pi_j^{\gamma+} &= \max\{(1 - \bar{a}_{k0} - \gamma \bar{a}_{i0})\bar{a}_{kj}, -(\bar{a}_{k0} + \gamma \bar{a}_{i0})\bar{a}_{kj} - \gamma \bar{a}_{ij}\} \\ &= -(\bar{a}_{k0} + \gamma \bar{a}_{i0})\bar{a}_{kj} + \max\{\bar{a}_{kj}, -\gamma \bar{a}_{ij}\} \\ \pi_0^{\gamma+} &= (1 - \bar{a}_{k0} - \gamma \bar{a}_{i0})\bar{a}_{k0} \end{aligned}$$

$$\gamma < 0 : \quad \begin{aligned} \pi_j^{\gamma-} &= \max\{(1 - \bar{a}_{k0} - \gamma \bar{a}_{i0})\bar{a}_{kj} + \gamma \bar{a}_{ij}, -(\bar{a}_{k0} + \gamma \bar{a}_{i0})\bar{a}_{kj}\} \\ &= -(\bar{a}_{k0} + \gamma \bar{a}_{i0})\bar{a}_{kj} + \max\{\bar{a}_{kj} + \gamma \bar{a}_{ij}, 0\} \\ \pi_0^{\gamma-} &= (1 - \bar{a}_{k0})(\bar{a}_{k0} + \gamma \bar{a}_{i0}) \end{aligned}$$

Using $\pi^{\gamma+}\bar{s}_J - \pi_0^{\gamma+}$ and $\pi^{\gamma-}\bar{s}_J - \pi_0^{\gamma-}$ in place of $\pi_i^\gamma \bar{x}_i + \pi^\gamma \bar{s}_J - \pi_0^\gamma$ in (31), the objective function of $(\mathrm{CGLP})_k$ corresponding to these cuts becomes

$$\frac{\pi^{\gamma+}\bar{s}_J - \pi_0^{\gamma+}}{1 + |\gamma| + \sum_{j \in J} |\bar{a}_{kj} + \gamma \bar{a}_{ij}|}$$

and

$$\frac{\pi^{\gamma-}\bar{s}_J - \pi_0^{\gamma-}}{1 + |\gamma| + \sum_{j \in J} |\bar{a}_{kj} + \gamma \bar{a}_{ij}|}$$

If we insert the expressions for $\pi^{\gamma+}$, $\pi_0^{\gamma+}$, $\pi^{\gamma-}$ and $\pi_0^{\gamma-}$ into the above, we obtain $f^+(\gamma)$ and $f^-(\gamma)$ respectively, as stated in the theorem. □

Next we sketch an algorithm for finding an optimal lift-and-project cut by pivoting in the simplex tableau of (LP).

Step 0. Solve (LP). Let $\bar{x}$ be an optimal solution and let k be such that $0 < \bar{x}_k < 1$.

Step 1. Let J index the nonbasic variables in the current basis. Compute the reduced costs r_{u_i} with $M_1 = \{j \in J : \bar{a}_{kj} < 0 \vee (\bar{a}_{kj} = 0 \wedge \bar{a}_{ij} > 0)\}$, $M_2 = J \setminus M_1$, and r_{v_i} with $M_1 = \{j \in J : \bar{a}_{kj} < 0 \vee (\bar{a}_{kj} = 0 \wedge \bar{a}_{ij} < 0)\}$, $M_2 = J \setminus M_1$ of u_i, v_i, corresponding to each row $i \neq k$ of the simplex tableau of (LP) according to (25).

Step 2. Let i_* be a row with $r_{u_{i_*}} < 0$ or $r_{v_{i_*}} < 0$. If no such row exists, go to Step 5.

Step 3. Identify the most improving pivot column j_* in row i_* by minimizing $f^+(\gamma_j)$ over all $j \in J$ with $\gamma_j > 0$ and $f^-(\gamma_j)$ over all $j \in J$ with $\gamma_j < 0$ and choosing the more negative of these two values.

Step 4. Pivot on $\bar{a}_{i_* j_*}$ and go to Step 1.

Step 5. If row k has no 0 entries, stop. Otherwise perturb row k by replacing every 0 entry by ε^t for some small ε and $t = 1, 2, \ldots$ (different for each entry). Go to step 1.

At termination, the simple disjunctive cut from row k (derived after removing the ε-perturbation, in case it is present) is an optimal lift-and-project cut; the mixed-integer Gomory cut from row k is an optimal strengthened lift-and-project cut.

When we compute the reduced costs in Step 1, we create a partition (M_1, M_2) of J according to Theorem 4B. When $\bar{a}_{kj} = 0$ for some $j \in J$ we are free to choose whether to assign j to M_1 or M_2. By assigning such a j to M_1 if $\bar{a}_{ij} > 0$ and to M_2 otherwise in the case of r_{u_i}, we make sure that the corresponding (CGLP) basis permits a nondegenerate pivot in column u_i. Thus, if $r_{u_i} < 0$, this pivot improves the solution. This is because with such a choice of M_1 and M_2 it is possible to increase u_i by a small amount in (27) without driving any of the u_j and v_j negative. Equivalently for r_{v_i}. Hence, when step 2 does not find a negative reduced cost, there is no improving pivot in the simplex tableau of (LP).

In order to explain the role of the perturbation (Step 5), we need to examine in more detail the connection between pivots in the (LP) simplex tableau (the small tableau) and pivots in the simplex tableau of (CGLP) (the large tableau). The set J defines a unique basis B of the small tableau, which corresponds to as many bases of the large tableau as there are partitions (M_1, M_2) of J satisfying the requirements of feasibility (i.e. $j \in M_1$ if $\bar{a}_{kj} < 0$ and $j \in M_2$ if $\bar{a}_{kj} > 0$). Now let us assume a certain partition (M_1, M_2) satisfying these requirements, corresponding to a feasible basis $\widetilde{B}$ of the large tableau. A pivot in row i of the small tableau replaces B with an adjacent basis B', but it may change the signs of many entries of row k, resulting in a change of the set of candidates for inclusion into M_1 or M_2. Thus any of the partitions (M_1', M_2') available after such a pivot may differ from the earlier partition (M_1, M_2) by several elements, which means that the basis $\widetilde{B}'$ of (CGLP) corresponding to the chosen partition (M_1', M_2') will differ from the earlier basis $\widetilde{B}$ by several columns, i.e. would be obtainable from $\widetilde{B}$ through several pivots in (CGLP).

When the algorithm comes to a point where Step 2 finds no row $i \neq k$ with $r_{u_i} < 0$ or $r_{v_i} < 0$, i.e. all the reduced costs of the large tableau are nonnegative, we could conclude that the solution is optimal if the reduced costs had all been calculated with respect to the same basis of the large tableau, i.e. with respect to the same partition (M_1, M_2). However, this is not the case, since the attempt to find a pivot that improves the cut from row k as much as possible makes us use a different partition (M_1, M_2) for every row i, as explained above. While in the absence of 0 entries in row k the partition (M_1, M_2) is unique (the same for all i), the presence of 0's in row k allows us to use different partitions for different rows, thereby gaining in efficiency. When all the reduced costs calculated in this way are nonnegative, then in order to make sure that the cut is optimal, we must recalculate the reduced costs from a unique basis of the large tableau, i.e. a unique partition (M_1, M_2). This is what the perturbation in Step 5 accomplishes, by eliminating the 0 entries in row k.

Since the perturbation is cumbersome and slows down the algorithm, in practice we run it without step 5, stopping when all reduced costs are nonnegative. Experience shows that the cuts obtained this way are on the average of roughly the same strength as those obtained by solving explicitly the (CGLP) (see section 9 for details).

9. Preliminary computational experience

We have implemented a version of this algorithm and compared its performance on 20 MIPLIB [8] problems with the standard way of optimizing the cut generating linear program. The problems were chosen as the subset of problems from [5] for which each individual (CGLP) can be solved within 2000 iterations, and range in size from 20×27

(bm21) to 728 × 2681 (p2756). All tests were run on a Sun Ultra 60 using a 360 MHz UltraSPARC-II processor with 512MB memory. CPLEX version 6.5 was used to solve the linear programs and also to preprocess the problems before applying the two cut generation procedures.

What we compared was a single round of cuts generated at the root node, that is one cut for every 0-1 variable fractional at the LP optimum. Table 1 shows the outcome of the comparison in terms of total number of pivots, total time and lower bound obtained. The numbers in the LP columns refer to the algorithm described here, which works with the simplex tableau of (LP) and never uses (CGLP) explicitly. The algorithm was implemented in C and does not use any other code. The numbers in the CGLP columns refer to the standard way of generating lift-and-project cuts, which formulates a higher dimensional cut generating linear program, $(\mathrm{CGLP})_k$, for each k such that $\bar{x}_k$ is fractional. Each of these linear programs is solved by calling CPLEX with a starting basis equivalent to the simple disjunctive cut from the optimal simplex tableau, as given by the relation (18).

As can be seen, working in the (LP) simplex tableau requires on the average 7–8 times fewer pivots to find an optimal cut, than doing the same thing in the (CGLP) tableau. Although we are comparing total time spent in our own experimental code against the highly efficient CPLEX, we observe on most of the problems a significant reduction in time. Finally, although the cuts generated by the two procedures are mostly the same, sometimes there are minor differences due to choices made under conditions of degener-

Table 1. LP versus CGLP: number of pivots, time, final cut strength

Problem	Preprocessed LP optimum	No. of cuts	No. of pivots		Time (sec)		LP optimum after cuts	
			LP	CGLP	LP	CGLP	LP	CGLP
bm23	20.57	6	20	36	0.00	0.02	22.11	22.11
egout	242.91	8	20	94	0.01	0.04	260.95	260.95
fxch.3	152.01	15	52	1382	0.10	0.46	157.96	159.04
genova6xs	10214	43	4621	37794	152.04	87.05	10220	10220
lseu	947.96	6	14	125	0.01	0.07	995.24	995.58
misc05	1230.90	11	130	836	0.35	0.49	1238.85	1232.45
mod008	290.93	5	27	468	0.05	0.68	291.08	291.07
p0033	2819.36	9	22	11	0.00	0.03	2844.98	2844.98
p0201	7125.00	22	127	2675	0.43	1.27	7135.00	7125.00
p0282	180000	24	35	435	0.09	0.48	219379	219379
p0291	2925.81	7	20	157	0.01	0.08	4928.6	5013.84
p0548	3126.38	45	15	1994	0.15	2.06	5630.77	5630.77
p2756	2702.39	63	55	12231	1.96	60.55	2933.11	2915.78
rgn	48.80	18	18	435	0.03	0.34	50.81	48.80
set1al	9784.13	197	1	788	0.55	6.65	13966.50	13953.00
stein45	15.33	45	830	4783	3.88	1.48	18.83	18.99
tsp43	5611.00	18	111	10172	3.56	27.53	5612.00	5611.00
utrans.2	207.74	14	81	951	0.12	0.37	215.45	215.46
utrans.3	267.32	19	93	1072	0.15	0.52	276.99	277.57
vpm1	16.43	16	63	1379	0.08	0.43	16.65	16.66

acy that are not always identical in the two procedures. The last two columns of Table 1 are meant to reflect these occasional differences. They show that while working in the (LP) tableau requires many times fewer pivots, the overall strength of the cuts generated with this procedure is on the average roughly the same as with the other one. The LP optimum after adding the cuts, while mostly equal for the two procedures, is slightly higher (meaning stronger cuts) when working in the (LP) tableau for 7 instances, and slightly higher when working in the (CGLP) tableau for 7 instances. All in all, these differences are insignificant.

10. Using lift-and-project to strengthen mixed integer Gomory cuts

The algorithm of section 9 for finding an optimal lift-and-project cut through a sequence of pivots in the simplex tableau of (LP) can also be interpreted as an algorithm for strengthening (improving) through a sequence of pivots a mixed integer Gomory cut derived from a row of the (LP) simplex tableau. The first pivot in this sequence results in the replacement of the mixed integer Gomory (MIG) cut from the row associated with x_k in the optimal simplex tableau of (LP) (briefly row k) with the MIG cut from the same row k of another simplex tableau (not necessarily feasible), the one resulting from the pivot. The new cut is guaranteed to be more violated by the optimal LP solution $\bar{x}$ than was the previous cut. Each subsequent pivot results again in the replacement of the MIG cut from row k of the current tableau with a MIG cut from row k of a new tableau, with a guaranteed improvement of the cut. This algorithm is essentially an exact version of the heuristic procedure for improving mixed integer Gomory cuts described in [2] (Example 3.1, p. 9–11).

The nature of this improvement is best understood by viewing the MIG cut as a simple disjunctive cut, and considering the strengthening of the disjunction – a dichotomy between two inequalities – through the addition of multiples of other inequalities to either term, before actually taking the disjunction.

Here is a brief illustration of what this strengthening procedure means, on an example small enough for the purpose, yet hard enough for standard cuts, the Steiner triple problem with 15 variables and 35 constraints (problem stein15 of [8]). To mitigate the effects of symmetry, we replaced the objective function of $\sum_j x_j$ by $\sum_j jx_j$.

The linear programming optimum is

$$\bar{x} = (1, 1, 1, 1, 1, 0.5, 0.5, 0.5, 0.5, 0.5, 0, 0, 0, 0, 0)$$

with a value of 35. The integer optimum is

$$x^* = (1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0),$$

with value 45.

Generating one mixed integer Gomory cut from each of the five fractional variables and solving the resulting linear program yields a solution x^1 of value 39. Iterating this procedure 10 times, each time generating one MIG cut from each of the fractional variables of the current solution and solving the resulting linear program, yields the solution

$$x^{10} = (0.97, 1, 1, 0.93, 1, 0.73, 0.64, 0.33, 0.61, 0.86, 0.67, 0, 0.35, 0.55, 0.32)$$

with a value of 42.73.

But if instead of using the five MIG cuts as they are, we first improve them by our pivoting algorithm, then use the five improved cuts in place of the original ones, we get a solution $\tilde{x}^1$ of value 41.41. If we then iterate this procedure 10 times, using every time the improved cuts in pace of the original ones, we obtain the solution

$$\tilde{x}^{10} = (1, 1, 1, 1, 1, 1, 1, 0.94, 0.72, 0.29, 0, 0, 0, 0, 0)$$

with a value of 44.85.

The difference between x^{10} and $\tilde{x}^{10}$ is striking. However, even more striking are the details. Lack of space limits us to discussing the first out of the ten iterations of the above procedure. Here is how the improving pivots affect the amount of violation, defined as $\beta - \alpha\bar{x}$ for the cut $\alpha x \geq \beta$ normalized as in section 3, and the distance, meaning the Euclidean distance of $\bar{x}$ from the cut hyperplane:

	Violation	Distance
Cut from x_6: original MIG	0.0441	0.1443
optimal (after 3 pivots)	0.0833	0.2835
Cut from x_7: original MIG	0.0625	0.1768
optimal (after 1 pivot)	0.0714	0.2085
Cut from x_8: original MIG	0.0577	0.2023
optimal (after 1 pivot)	0.0833	0.2835
Cut from x_9: original MIG	0.0500	0.1744
optimal (after 3 pivots)	0.0833	0.2887
Cut from x_{10}: original MIG	0.0500	0.1744
optimal (after 4 pivots)	0.0833	0.2835

In the process of strengthening the 5 MIG cuts in the first iteration, 12 new cuts are generated. If, instead of replacing the original MIG cuts with the improved ones, we keep all the cuts generated and solve the problem with all 17 cuts (the 5 initial ones plus the 12 improved ones), we get exactly the same solution $\tilde{x}^1$ as with the 5 final improved cuts only: the original MIG cuts as well as the intermediate cuts resulting from the improving pivots (except for the last one) are made redundant by the 5 final improved cuts.

A similar behavior is exhibited on the problems stein27 (with 27 variables and 117 constraints) and stein45 (45 variables, 330 constraints).

Since the improved MIG cuts resulting from our algorithm are equivalent to the (strengthened) optimal lift-and-project cuts, these findings corroborate those of [4]. The practical question that remains to be answered, is the following: does the gain in the quality of the cuts justify the computational effort for improving them? This can only be established experimentally, and it pretty much defines the next task in this area of research.

One last comment. Since the algorithm described here starts with a MIG cut from the optimal simplex tableau and stops with a MIG cut from another (usually infeasible) simplex tableau, one may ask what is the role of lift-and-project theory in this process? The answer is that it provides the guidance needed to get from the first MIG cut to the final one. It provides the tools, in the form of the reduced costs from the (CGLP) tableau of the auxiliary variables u_i and v_i, for identifying a pivot that is guaranteed to improve the cut, if one exists. Over the last three and a half decades there have been numerous attempts to improve mixed integer Gomory cuts by deriving them from tableau rows combined in different ways, but none of these attempts has succeeded in defining a procedure that is *guaranteed* to find an improved cut when one exists. The lift-and-project approach has done just that.

References

[1] E. Balas, "Intersection Cuts – A New Type of Cutting Planes for Integer Programming," *Operations Research, 19*, 1971, 19–39.

[2] E. Balas, "Disjunctive Programming," *Annals of Discrete Mathematics, 5*, 1979, 3–51.

[3] E. Balas, "Disjunctive Programming: Properties of the Convex Hull of Feasible Points." Invited paper in *Discrete Applied Mathematics, 89,* 1998, 1–44.

[4] E. Balas, S. Ceria and G. Cornuéjols, "A Lift-and-Project Cutting Plane Algorithm for Mixed 0-1 Programs," *Mathematical Programming, 58*, 1993, 295–324.

[5] E. Balas, S. Ceria and G. Cornuéjols, "Mixed 0-1 Programming by Lift-and-Project in a Branch-and-Cut Framework," *Management Science, 42,* 1996, 1229–1246.

[6] E. Balas and R. Jeroslow, "Strengthening Cuts for Mixed Integer Programs," *European Journal of Operations Research, 4*, 1980, 224–234.

[7] E. Balas and M. Perregaard, "Lift and Project for Mixed 0-1 Programming: Recent Progress," MSRR No. 627, September 1999.

[8] R.E. Bixby, S. Ceria, C.M. McZeal, M.W.P. Savelsbergh, "An updated Mixed Integer Programming Library: MIPLIB 3.0",
`http://www.caam.rice.edu/~bixby/miplib/miplib.html`.

[9] G. Cornuéjols, Y. Li, "On the Rank of Mixed 0,1 Polyhedra", in K. Aardal et al. (editors), *Integer Programming and Combinatorial Optimization* Proceedings of IPCO 8, *Lecture Notes in Computer Science, 2081,* 2001, 71–77.

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

[11] F. Eisenbrand and A. Schulz, "Bounds on the Chvatal rank of polytopes in the 0-1 cube," in G. Cornuéjols et al. (editors), *Integer Programming and Combinatorial Optimization,* Proceedings of IPCO 7, *Lecture Notes in Computer Science, 1610,* 1999, 137–150.

[12] R. Gomory, "Outline of an Algorithm for Integer Solutions to Linear Programs." *Bulletin of the American Mathematical Society, 64,* 1958, 275–278.

[13] R. Gomory, "An Algorithm for the Mixed Integer Problem." Technical Report RM-2597, The RAND Corporation, 1960.

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

Copyright © 2003 EBSCO Publishing