Lift-and-project for general two-term disjunctions

Tamás Kis*

February 7, 2014

Abstract

In this paper we generalize the cut strengthening method of Balas and Perregaard for 0/1 mixed-integer programming to disjunctive programs with general two-term disjunctions. We apply our results to linear programs with complementarity constraints.

Keywords Disjunctive programming, Lift-and-Project cuts.

1 Introduction

Consider the disjunctive program

$$(DP)\qquad \begin{aligned} &\min cx &(1)\\ &\text{s.t.}\\ &Ax \geq b &(2)\\ &x \geq 0 &(3)\\ &d^{k,1}x \geq d_0^{k,1} \ \vee\ d^{k,2}x \geq d_0^{k,2}, \quad k \in \Gamma &(4) \end{aligned}$$

where Γ is a finite set of indices. Constraints (4) are called *disjunctive*, a term coined by Egon Balas [2]. Notice that (4) consists of *two-term* disjunctions. A vector $\hat{x}$ is a *feasible solution* of the system (2)-(4) if it satisfies the constraints (2)-(3), and for each $k \in \Gamma$ at least one of the two terms from (4). A mathematical program with disjunctive constraints is called *disjunctive program* (DP) in [2].

In this paper we will focus on strengthening the LP relaxation of disjunctive programs by cutting planes. The LP relaxation of (1)-(4) is obtained by dropping the disjunctive constraints (4). Balas [5] gave a primal and a dual description of the convex hull of all points in the set $P \cap (\bigvee_{h \in Q} D^h x \geq d_0^h)$, where P is a convex polyhedron, and Q is a finite set of indices. The primal description was merely a linear program which consisted of $|Q|$ copies of the linear description of P, and the dual description was a characterization of all the valid inequalities and facets of the convex hull of feasible points, which was shown to be polyhedral.

A *disjunctive cut* for (DP) is any inequality valid for $\overline{\text{conv}}(P^{k,1} \cup P^{k,2})$ (the closed convex hull of $P^{k,1} \cup P^{k,2}$), where $P^{k,t} = \{x \mid Ax \geq b,\ d^{k,t}x \geq d_0^{k,t}\}$,

*Address: Computing and Automation Research Institute, Kende str. 13-17, H1111 Budapest, Hungary, e-mail: kis.tamas@sztaki.mta.hu

$t = 1, 2$. Finding good disjunctive cuts is not an easy task. The lift-and-project procedure was developed by Balas et al [3],[4] to strengthen Gomory cuts for programs with 0-1 variables. Gomory cuts are generated from disjunctions of the form $\pi x \leq \pi_0 \vee \pi x \geq \pi_0 + 1$ (*split disjunctions*), where $(\pi, \pi_0) \in \mathbb{Z}^{n_1} \times \mathbb{Z}$, and x is the set of integer variables of a mixed-integer linear program

$$\min\{cx + gy \mid (x, y) \in P,\ x \text{ integer}\}, \tag{5}$$

where $P = \{(x, y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \mid Ax + Gy \geq b,\ x \in [0, 1]^{n_1}, 0 \leq y \leq u\}$ is a convex polyhedron. The linear relaxation of (5) is obtained by dropping the condition "x integer". In [3] and [4], cuts were generated from disjunctions of the form $x_i \leq 0 \vee x_i \geq 1$ by solving a *Cut Generating Linear Program* (CGLP):

$$
\begin{aligned}
&\min \alpha(\hat{x}, \hat{y}) - \beta && (6)\\
&\text{s.t.} && \\
&\alpha - u^T(A, G) + u_0 e_i \geq 0 && (7)\\
&\alpha - v^T(A, G) - v_0 e_i \geq 0 && (8)\\
&-\beta + u^T b \geq 0 && (9)\\
&-\beta + v^T b + v_0 \geq 0 && (10)\\
&\alpha \text{ normalization, or } \beta \text{ normalization} && (11)\\
&u, u_0, v, v_0 \geq 0 && (12)
\end{aligned}
$$

In the objective function, $(\hat{x}, \hat{y})$ is a basic solution of the linear relaxation of (5) with $\hat{x}_i$ fractional, which we want to cut off. Constraints (7)-(10) express that the inequality $\alpha x \geq \beta$ is valid for both of the polyhedra $P_0 = \{(x, y) \in P \mid -x_i \geq 0\}$, and $P_1 = \{(x, y) \in P \mid x_i \geq 1\}$. The α-normalization in (11) means $\sum_j |\alpha_j| = 1$, and the β-normalization means either $\beta = 1$ or $\beta = -1$, and it is needed to ensure a finite optimum value. There are other normalizations possible, such as $u^T \mathbb{1} + u_0 + v^T \mathbb{1} + v_0 = 1$, where $\mathbb{1}$ is the vector of all ones of appropriate dimension, which is used in Balas and Perregaard [8]. A drawback of the original approach was the computational overhead caused by forming CGLP, even in a reduced space, and by the many pivots needed to solve it to optimality. Both of these problems were eliminated by Balas and Perregaard [8], and Perregaard [15] by establishing a direct correspondence between the bases of the linear relaxation of (5) and the basic solutions of CGLP (6)-(12), and by showing how to pivot in the simplex tableau of (5) (*small tableau*) in order to improve the objective function value of CGLP. They have also shown that by pivoting in the small tableau, the true optimum of CGLP can be found. There are a number of other papers dealing with strengthening split disjunctions, see e.g., [6], [7], [9], [11], and also multi-term disjunctions [14]. Before presenting our results, we need the following:

Definition 1 *Let $P_1, P_2 \subset \mathbb{R}^n$ be polyhedra with $P_1 \subset P_2$ (P_2 is a relaxation of P_1).*

- *We say that $\alpha x \geq \beta$ is* valid *for P_1 if and only if $P_1 \subseteq \{x \in \mathbb{R}^n : \alpha x \geq \beta\}$.*
- *Let $\alpha_1 x \geq \beta_1$ and $\alpha_2 x \geq \beta_2$ be valid inequalities for P_1. We say that $\alpha_1 x \geq \beta_1$* dominates *$\alpha_2 x \geq \beta_2$ on P_2 if and only if $\{x \in P_2 : \alpha_1 x \geq \beta_1\} \subseteq \{x \in P_2 : \alpha_2 x \geq \beta_2\}$.*

- *Let S and R be sets of valid inequalities for P_1. We say that S* dominates *R on P_2, if any inequality of R is equal to or dominated by a convex combination of inequalities in S. The domination is* strict *if S dominates R, but R does not dominate S, i.e., there exists an inequality in S which is not equal to or dominated by a convex combination of inequalities in R.*

Main Results of this paper. We generalize the Lift-and-Project procedure of Balas and Perregaard [8] to general two-term disjunctions. The Cut Generating Linear Program is very similar to that of integer programming, where the disjunctions take the form $\pi x \leq \pi_0 \vee \pi x \geq \pi_0 + 1$ in general. We are interested in the more general case when the two hyperplanes corresponding to the two terms are not parallel. We will discuss the similarities and point out the differences to the case of split disjunctions. One of our main findings is that the Cut Generating Linear Program (CGLP) has basic solutions that, unlike in case of split disjunctions, cannot be linked to bases of the small tableau. As a consequence, in case of general two term disjunctions, the set of Lift-and-Project cuts may strictly dominate on the LP relaxation of (DP) the set of disjunctive cuts that can be derived from the bases of the simplex tableau by a standard formula. The heart of the Lift-and-Project procedure of Balas and Perregaard is that it replaces pivot sequences in CGLP by single pivots in the small tableau. We will show that the same technique, with appropriate modifications, works in the general two-term case, but unlike in the case of split disjunctions, it is not guaranteed that an optimal solution can be reached by a sequence of pivots in the small tableau. While describing the method, we provide a new derivation of reduced costs needed for identifying pivot rows. We will also evaluate the Cut-generation method on some benchmark instances, the main goal being to compare the Cut-generation procedure in the small tableau to solving CGLP by a linear programming solver.

The structure of the paper. In the next section, we recapitulate how to generate disjunctive cuts from the simplex tableau along with the concept of Cut-Generating Linear Programs (CGLP), and that of Lift-and-Project cuts. In Section 3 we prove that any basis of the LP relaxation induces a disjunctive cut and also an equivalent solution of CGLP. In Section 4 we generalize the cut-generation procedure of [8] to disjunctive programs with general two-term disjunctions, and also show its limitations. A sufficient condition of optimal termination is given in Section 5. In Section 6, we evaluate the cut-generation procedure on benchmark problems from the literature.

Notation and terminology. Throughout the paper, A is a matrix in $\mathbb{R}^{m \times n}$, $\tilde{A} = [A^T, I]^T$, i.e., we add to the rows of A the identity matrix, and $\tilde{b} = [b^T, 0]^T$ is the corresponding right-hand-side of the system $\tilde{A}x \geq \tilde{b}$, which is equivalent to $Ax \geq b, x \geq 0$. The linear inequality system $Ax \geq b$, can be extended to a system of equations by new surplus variables $x_{n+1}, \ldots, x_{n+m}$, i.e., $[A, -I]x = b$. If $B \subseteq \{1, \ldots, n\}$, then the submatrix of A with columns indexed by B is A^B, whereas if $Q \subseteq \{1, \ldots, m\}$, the submatrix of A consisting of those rows indexed by Q is A_Q. Hence, A_Q^B is a submatrix of A with columns in B and rows in Q. A *basis* of $[A, -I]$ is a set of m linearly independent columns $B \subseteq \{1, \ldots, n+m\}$. Let $N = \{1, \ldots, n+m\} \setminus B$ be the set of nonbasic variables. Clearly, $|N| = n$, the number of columns of A. Let $P_1, P_2 \subseteq \mathbf{R}_+^n$ be convex polyhedra in the nonnegative orthant. The *convex hull of the union* of P_1 and P_2 is a convex set, which may not be a polyhedron as it is not necessarily closed if P_1 or P_2

is unbounded, see Balas [5]. Therefore, throughout this paper, we will take the closure of conv$(P_1 \cup P_2)$, which is also called the *closed convex hull* of $P_1 \cup P_2$, denoted by $\overline{\text{conv}}(P_1 \cup P_2)$. We can obtain valid inequalities for $\overline{\text{conv}}(P_1 \cup P_2)$ from valid inequalities for P_1 and P_2 using the *disjunctive principle*, see Balas [2], [5]. Let $\pi^i x \geq \pi^i_0$ be a valid inequality for P_i, $i = 1, 2$. Since both polyhedra are in the nonnegative orthant, the inequality

$$\sum_{j=1}^{n} \max\{\pi^1_j, \pi^2_j\} x_j \geq \min\{\pi^1_0, \pi^2_0\}$$

is valid for $\overline{\text{conv}}(P_1 \cup P_2)$.

2 Preliminaries

In this section we recapitulate known techniques for generating cuts for disjunctive programs.

2.1 Generation of disjunctive cuts from the simplex tableau

Given a solution $\hat{x}$ of the LP relaxation of (DP) which violates some of the disjunctions in (4), i.e., $d^{k,1}\hat{x} < d^{k,1}_0$ and $d^{k,2}\hat{x} < d^{i,2}_0$. A valid inequality for (DP) can be generated from the basic solution $\hat{x}$ as a disjunctive cut [2]. Firstly, we need to express both inequalities $d^{k,t}x \geq d^{k,t}_0$, $t = 1, 2$, in the basis of (DP) corresponding to $\hat{x}$. The simplex tableau in basis B is

$$x_i + \sum_{j \in N} \bar{a}_{ij} x_j = \bar{a}_{i0}, \quad i \in B \tag{13}$$

where $\bar{a}_{ij} = [(A^B)^{-1} A^N]_{ij}$ with $A = [A^B, A^N]$. In order to express a new inequality $d^{k,t}x \geq d^{k,t}_0$ in basis B, we can use a result of Balas et al. [10].

Proposition 1 *Suppose $dx \geq d_0$ is violated in the basic solution $\hat{x}$ corresponding to a basis B of the linear system $Ax = b$, and let $s = dx - d_0$ be a new surplus variable. Then*

$$s + (d^B \bar{A}^N - d^N) x_N = d^B \bar{a}_0 - d_0 \tag{14}$$

Proof Firstly, we partition the variables to basic and nonbasic, x_B and x_N, and rewrite the system as follows:

$$\begin{array}{lcccl} A^B x_B & & & + \; A^N x_N & = b \\ d^B x_B & - & s & + \; d^N x_N & = d_0 \end{array} \tag{15}$$

Since $\hat{x}$ violates $dx \geq d_0$, s is basic (takes a negative value). The submatrix corresponding to the basis $B \cup \{s\}$, and its inverse is

$$\begin{pmatrix} A^B & 0 \\ d^B & -1 \end{pmatrix}^{-1} = \begin{pmatrix} (A^B)^{-1} & 0 \\ d^B (A^B)^{-1} & -1 \end{pmatrix}.$$

Multiplying (15) by the basis inverse from the left we get

$$\begin{array}{rcrl} x_B & + & \bar{A}^N x_N & = \bar{a}_0 \\ s & + & (d^B \bar{A}^N - d^N) x_N & = d^B \bar{a}_0 - d_0 \end{array}$$

The statement follows. □

Let $s^{k,1}$ and $s^{k,2}$ be the basic surplus variables in $d^{k,t}x - s^{k,t} = d_0^{k,t}$, for $t = 1, 2$, in a basis B determining $\hat{x}$ (recall that both of these inequalities are violated by $\hat{x}$ by assumption). Using Proposition 1, we immediately get that in the basis B, the two violated inequalities can be expressed as

$$s^{k,1} + ((d^{k,1})^B \bar{A}^N - (d^{k,1})^N) x_N = (d^{k,1})^B \bar{a}_0 - d_0^{k,1}$$
$$s^{k,2} + ((d^{k,2})^B \bar{A}^N - (d^{k,2})^N) x_N = (d^{k,2})^B \bar{a}_0 - d_0^{k,2}$$

Now we can equivalently express the disjunction as $s^{k,1} \geq 0 \vee s^{k,2} \geq 0$. Hence, using the disjunctive principle, we obtain the following *disjunctive cut from the basis of the simplex tableau*

$$\sum_{j \in N} \max\{\pi_j^1, \pi_j^2\} x_j \geq \pi_0, \tag{16}$$

where $\pi_j^1 := \bar{d}_j^{k,1} \bar{d}_0^{k,2}$, $\pi_j^2 := \bar{d}_j^{k,2} \bar{d}_0^{k,1}$, $j \in N$, $\pi_0 := \bar{d}_0^{k,1} \bar{d}_0^{k,2}$, and $\bar{d}^{k,t} := ((d^{k,t})^B \bar{A}^N - (d^{k,t})^N)$, $\bar{d}_0^{k,t} = (d^{k,t})^B \bar{a}_0 - d_0^{k,t}$, for $t = 1, 2$. Notice that since both of the two terms $d^{k,t}x \geq d_0^{k,t}$, $t = 1, 2$, are violated, $s^{k,t} = \bar{d}_0^{k,t} < 0$ for $t = 1, 2$. Therefore, the right-hand-side of (16) is positive. Therefore, $\hat{x}$ is cut off by (16), since $\hat{x}_j = 0$ for $j \in N$.

Clearly, the cuts (16) are valid for $\overline{\text{conv}}(P^{k,1} \cup P^{k,2})$ (the closed convex hull of $P^{k,1} \cup P^{k,2}$), this is why we call them disjunctive cuts (cf. Section 1). However, they are derived by a particular formula from a basis of the simplex tableau, and as we will see in the next section, there are other ways of getting cuts valid for $\overline{\text{conv}}(P^{k,1} \cup P^{k,2})$.

2.2 Lift-and-project cuts

To find a violated inequality for a disjunctive program, Balas [5] and Balas et al. [3] propose to solve a linear program. Suppose $\hat{x}$ violates a disjunction in (4), the *Cut Generating Linear Program* is

$$(CGLP)_k \qquad \begin{aligned} \min_{(\alpha,\beta,u,v,u_0,v_0)} \quad & \alpha\hat{x} - \beta && (17)\\ \text{s.t.} \quad & \alpha - u\tilde{A} - u_0 d^{k,1} = 0, && (18)\\ & \alpha - v\tilde{A} - v_0 d^{k,2} = 0, && (19)\\ & \beta - u\tilde{b} - u_0 d^{k,1} = 0, && (20)\\ & \beta - v\tilde{b} - v_0 d^{k,2} = 0, && (21)\\ & u\mathbb{1} + v\mathbb{1} + u_0 + v_0 = 1, && (22)\\ & u, v, u_0, v_0 \geq 0. \end{aligned}$$

Notice that $\tilde{A} = [A^T, I]^T$. Here, $\mathbb{1}$ denotes the $m + n$ dimensional column vector of all ones. The cut sought is of the form $\alpha x \geq \beta$. The objective function prescribes the generation of a cut of maximum violation in the sense that the difference between the right and left hand sides with respect to the feasible solution $\hat{x}$ is the largest. Constraints (18) and (20) ensure that $\alpha x \geq \beta$ is valid for the polyhedron $P^{k,1} = \{x \mid Ax \geq b, x \geq 0, d^{k,1}x \geq d_0^{k,1}\}$, whereas (19) and (21) yields that $\alpha x \geq \beta$ is also valid for the polyhedron $P^{k,2} = \{x \mid Ax \geq b, x \geq 0, d^{k,2}x \geq d_0^{k,2}\}$. Since these inequalities define a cone, we need the normalization constraint (22), otherwise the optimum value may be unbounded. The cuts generated by $(CGLP)_k$ are called *Lift-and-Project cuts*. The results of Balas [5] imply that

Proposition 2 *The optimum value of* $(CGLP)_k$ *is negative if and only if* $\hat{x}$ *is not in the set* $\overline{\text{conv}}(P^{k,1} \cup P^{k,2})$.

Notice that $(CGLP)_k$ is a generalization of that of Balas and Perregaard [8] (6)-(12) for mixed 0/1 programming. By generalising the results of [8], we obtain the following:

Proposition 3 *Unless* $u_0 > 0$ *and* $v_0 > 0$*, the optimum value of* $(CGLP)_k$ *is non-negative.*

By substitution for α and β, we obtain an equivalent linear program

$$(CGLP)'_k \qquad \begin{aligned} \min_{(u,v,u_0,v_0)} \quad & (u\tilde{A} + u_0 d^{k,1})\hat{x} - u\tilde{b} - u_0 d_0^{k,1} \\ \text{s.t.} \quad & (u - v)\tilde{A} + u_0 d^{k,1} - v_0 d^{k,2} = 0, \\ & (u - v)\tilde{b} + u_0 d_0^{k,1} - v_0 d_0^{k,2} = 0, \\ & u\mathbb{1} + v\mathbb{1} + u_0 + v_0 = 1, \\ & u, v, u_0, v_0 \geq 0. \end{aligned} \tag{23}$$

A slightly different cut generation LP is developed by Andersen et al. [1] for disjunctive programming with two term disjunctions, where each term is a conjunction of inequalities.

3 Correspondence between disjunctive cuts and lift-and-project cuts

Now we establish a connection between the disjunctive cuts (16) and the lift-and-project cuts from $(CGLP)_k$. Let B and N be the set of basic and nonbasic variables in the simplex tableau (13). Denote $\tilde{A}_N$ the submatrix of $\tilde{A}$ with rows corresponding to nonbasic slack and structural variables. Since $|N| = n$, and $\tilde{A}$ has n columns, $\tilde{A}_N$ is a square submatrix of $\tilde{A}$. We will need the following results.

Lemma 1 *(Balas and Perregaard [8]) If* N *is the set of nonbasic variables in a simplex tableau of* $[A, -I]$*, then* $\tilde{A}_N$ *is invertible. Moreover, in the simplex tableau (13) the coefficients* $\bar{a}_{ij}$ *for* $i \in B$ *and* $j \in N$*, and the right hand sides* $\bar{a}_{i0}$ *for* $i \in B$ *satisfy*

$$\bar{a}_{ij} = -(\tilde{A}_i \tilde{A}_N^{-1})_j, \tag{24}$$

$$\bar{a}_{i0} = \tilde{A}_i \tilde{A}_N^{-1} \tilde{b}_N - \tilde{b}_i. \tag{25}$$

The following result is not explicitly stated in [8], but it can be easily read out from the proof of Lemma 8 in [8].

Lemma 2 *Let $B' \subset \{1,\ldots,n{+}m\}$ with $|B'| = m$, and $N' = \{1,\ldots,n{+}m\}\setminus B'$. Then B' is a basis of $[A,-I]$ if and only if $\tilde{A}_{N'}$ is nonsingular.*

Now we are ready to prove the following:

Theorem 1 *Suppose disjunction $d^{k,1}x \geq d_0^{k,1} \vee d^{k,2}x \geq d_0^{k,2}$ is violated by the basic solution $\hat{x}$ in a basis B of $[A,-I]$. Then, $(CGLP)_k$ admits a basic feasible solution $(\alpha,\beta,u,v,u_0,v_0)$ such that the disjunctive cut $\pi s_N \geq \pi_0$ from the basis B, and the lift-and-project cut $\alpha x \geq \beta$ are equivalent.*

Proof Let $\pi_j^1 := \bar{d}_j^{k,1}\bar{d}_0^{k,2}$, $\pi_j^2 := \bar{d}_j^{k,2}\bar{d}_0^{k,1}$, $\pi_j := \max\{\pi_j^1,\pi_j^2\}$ for $j \in N$, and $\pi_0 := \bar{d}_0^1\bar{d}_0^2$, where $N = \{1,\ldots,n{+}m\}\setminus B$ is the set of nonbasic variables. Notice that $\pi x_N \geq \pi_0$ is the disjunctive cut (16) determined from the simplex tableau.

We define the values of α,β,u,v,u_0,v_0 as follows: Let θ be a positive value to be fixed later, and

$$\begin{aligned} \theta\alpha &:= \pi\tilde{A}_N, & \theta\beta &:= \pi_0 + \pi\tilde{b}_N,\\ \theta u_N &:= \pi - \pi^1, & \theta v_N &= \pi - \pi^2,\\ \theta u_0 &:= -\bar{d}_0^{k,2}, & \theta v_0 &:= -\bar{d}_0^{k,1}. \end{aligned}$$

For all $j \in B$, $u_j = v_j = 0$ (the dual variables corresponding to the basic slack and structural variables of $\tilde{A}x \geq \tilde{b}$ get value 0). We verify that this particular choice of $(\alpha,\beta,u,v,u_0,v_0)$ is a feasible solution of $(CGLP)_k$.

$$\begin{aligned} \theta(\alpha - u_N\tilde{A}_N - u_0 d^{k,1}) &= \pi\tilde{A}_N - (\pi - \pi^1)\tilde{A}_N + \bar{d}_0^{k,2}d^{k,1}\\ &= \bar{d}_0^{k,2}(d^{k,1} - (d^{k,1}\tilde{A}_N^{-1})\tilde{A}_N) = 0 \end{aligned}$$

Here, we exploited that $\bar{d}^{k,1} = -(d^{k,1}\tilde{A}_N^{-1})$ by Lemma 1. One similarly shows that $\theta(\alpha - v_N\tilde{A}_N - v_0 d^{k,2}) = 0$. Furthermore, using (25) we obtain

$$\begin{aligned} \theta(\beta - u_N\tilde{b}_N - u_0 d_0^{k,1}) &= \pi_0 + \pi\tilde{b}_N - (\pi - \pi^1)\tilde{b}_N + \bar{d}_0^{k,2}d_0^{k,1}\\ &= \pi_0 + \bar{d}_0^{k,2}(d_0^{k,1} - (d^{k,1}\tilde{A}_N^{-1})\tilde{b}_N) = \pi_0 + \bar{d}_0^{k,2}(-\bar{d}_0^{k,1}) = 0. \end{aligned}$$

We can prove similarly that $\theta(\beta - v_N\tilde{b}_N + v_0 d^{i,2}) = 0$. Now we can define θ as

$$\theta := (\pi - \pi_N^1)\mathbb{1} + (\pi - \pi_N^2)\mathbb{1} - \bar{d}_0^{k,1} - \bar{d}_0^{k,2}.$$

Clearly, $\theta > 0$ and $u\mathbb{1} + v\mathbb{1} + u_0 + v_0 = 1$.

We verify that $u,v,u_0,v_0 \geq 0$. Since $\pi_j = \max\{\pi_j^1,\pi_j^2\}$ for $j \in N$, it follows that u_j and v_j are non-negative and at most one of them is greater than zero. Finally, since $\bar{d}_0^{k,1} < 0$ and $\bar{d}_0^{k,2} < 0$ by assumption, it follows that $u_0, v_0 > 0$.

To show that the solution $(\alpha,\beta,u,v,u_0,v_0)$ of $(CGLP)_k$ defined above is *basic*, firstly we define a partitioning of N into two subsets, M_1 and M_2 as follows. If $\pi_j^1 < \pi_j^2$, then $j \in M_1$; if $\pi_j^1 > \pi_j^2$, then $j \in M_2$, and if $\pi_j^1 = \pi_j^2$, break ties arbitrarily. Since $u_j = 0$ if $j \notin M_1$, and $v_j = 0$ if $j \notin M_2$, we know that $(u_{M_1}, v_{M_2}, u_0, v_0)$ satisfy the constraints

$$(u_{M_1}, -v_{M_2}) \begin{pmatrix} \tilde{A}_{M_1} \\ \tilde{A}_{M_2} \end{pmatrix} + u_0 d^{k,1} - v_0 d^{k,2} = 0, \tag{26}$$

$$(u_{M_1}, -v_{M_2}) \begin{pmatrix} \tilde{b}_{M_1} \\ \tilde{b}_{M_2} \end{pmatrix} + u_0 d_0^{k,1} - v_0 d_0^{k,2} = 0, \tag{27}$$

$$u_{M_1} \mathbb{1}_{M_1} + v_{M_2} \mathbb{1}_{M_2} + u_0 + v_0 = 1. \tag{28}$$

Since $\tilde{A}_N = \begin{pmatrix} \tilde{A}_{M_1} \\ \tilde{A}_{M_2} \end{pmatrix}$, and $\tilde{A}_N$ is invertible by Lemma 1, we have from (26):

$$(u_{M_1}, -v_{M_2}) = -u_0 d^{k,1} \tilde{A}_N^{-1} + v_0 d^{k,2} \tilde{A}_N^{-1}.$$

Using (24) we obtain

$$\begin{aligned} u_j &= u_0 \bar{d}_j^{k,1} - v_0 \bar{d}_j^{k,2}, \quad j \in M_1, \\ v_j &= -u_0 \bar{d}_j^{k,1} + v_0 \bar{d}_j^{k,2}, \quad j \in M_2. \end{aligned} \tag{29}$$

Moreover, from (27) it follows that

$$-u_0 d^{k,1} \tilde{A}_N^{-1} \tilde{b}_N + v_0 d^{k,2} \tilde{A}_N^{-1} \tilde{b}_N + u_0 d_0^{k,1} - v_0 d_0^{k,2} = 0.$$

Using (25) we get

$$-u_0 \bar{d}_0^{k,1} + v_0 \bar{d}_0^{k,2} = 0. \tag{30}$$

On the other hand, substituting (29) into (28) gives

$$u_0 (1 + \sum_{j \in M_1} \bar{d}_j^{k,1} - \sum_{j \in M_2} \bar{d}_j^{k,1}) + v_0 (1 - \sum_{j \in M_1} \bar{d}_j^{k,2} + \sum_{j \in M_2} \bar{d}_j^{k,2}) = 1. \tag{31}$$

The determinant of the linear system (30) and (31) over the variables u_0 and v_0 is

$$\delta = (-\bar{d}_0^{k,2})(1 + \sum_{j \in M_1} \bar{d}_j^{k,1} - \sum_{j \in M_2} \bar{d}_j^{k,1}) + (-\bar{d}_0^{k,1})(1 - \sum_{j \in M_1} \bar{d}_j^{k,2} + \sum_{j \in M_2} \bar{d}_j^{k,2}).$$

Observe that this quantity is precisely the value θ, defined above. To see this, using the definitions of π^1, π^2, M_1 and M_2 we obtain

$$\begin{aligned} \theta &= (\pi - \pi^1)\mathbb{1} + (\pi - \pi^2)\mathbb{1} - \bar{d}_0^{k,1} - \bar{d}_0^{k,2} \\ &= \sum_{j \in M_1} (\bar{d}_j^{k,2} \bar{d}_0^{i,1} - \bar{d}_j^{k,1} \bar{d}_0^{k,2}) + \sum_{j \in M_2} (\bar{d}_j^{k,1} \bar{d}_0^{k,2} - \bar{d}_j^{k,2} \bar{d}_0^{k,1}) - \bar{d}_0^{k,1} - \bar{d}_0^{k,2}. \end{aligned}$$

Rearranging terms gives the determinant δ. Since the determinant equals θ, a positive number, u_0 and v_0 are uniquely determined. Since w.l.o.g. α and β, are basic, u_{M_1} and v_{M_2} are uniquely determined by u_0 and v_0, and the rest of the variables are nonbasic, $(\alpha, \beta, u, v, u_0, v_0)$ is a basic solution.

Finally, the two cuts are equivalent, since

$$\theta(\alpha x - \beta) = \pi \tilde{A}_N x - (\pi_0 + \pi \tilde{b}_N) = \pi(\tilde{A}_N x - \tilde{b}_N) - \pi_0 = \pi s_N - \pi_0,$$

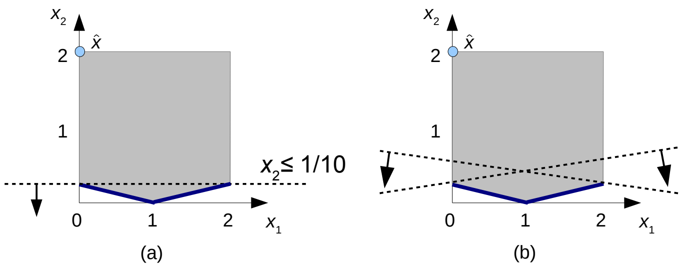

Figure 1: (a) The LP relaxation of (DP), and the optimal Lift-and-Project cut. (b) The two disjunctive cuts from two bases of the small tableau. The arrows in the figure point into to direction of points *valid* for the corresponding cut.

where x is the set of variables of the system $Ax \geq b$, and s_N is the set of nonbasic variables in the system $[A, -I]s = b$ $(s_N = \tilde{A}_N x - \tilde{b}_N)$. □

Unlike in the case of split disjunction, the converse of this theorem does not hold in general. The following theorem characterizes those bases of CGLP that give rise to lift-and-project cuts equivalent to disjunctive cuts from the small tableau.

Theorem 2 *Let $(\alpha, \beta, u, v, u_0, v_0)$ be a basic feasible solution of $(CGLP)_k$ with negative objective function value such that all the components of α, β, u_0, v_0, those of u indexed by M_1, and those of v indexed by M_2 are basic, where $M_1, M_2 \subset \{1, \ldots, n+m\}$ with $|M_1|+|M_2| = n$. If the square matrix $[(\tilde{A}_{M_1})^T, (\tilde{A}_{M_2})^T]$ is nonsingular, then $B' = \{1, \ldots, n+m\} \setminus (M_1 \cup M_2)$ is a basis of $[A, -I]$, and the disjunctive cut $\pi x \geq \pi_0$ from the disjunction $d^{k,1}x \geq d_0^{k,1} \vee d^{k,2}x \geq d_0^{k,2}$ derived using (16) from the simplex tableau corresponding to B', and the lift-and-project cut $\alpha x \geq \beta$ are equivalent.*

Theorem 2 is similar to Theorem 4A of [8], but there is a crucial difference. It has a condition, namely, the (square) submatrix $[(\tilde{A}_{M_1})^T, (\tilde{A}_{M_2})^T]$ of $\tilde{A}^T$ must be nonsingular. This need not be assumed if $(CGLP)_k$ depends on a split disjunction, because in that case, any feasible basis of $(CGLP)_k$ with negative objective function value automatically satisfies this condition. However, the proof parallels that of Theorem 4A of [8], once we know that $[(\tilde{A}_{M_1})^T, (\tilde{A}_{M_2})^T]$ is nonsingular, since this implies that $B' = \{1, \ldots, n+m\} \setminus (M_1 \cup M_2)$ is a basis of the small tableau by Lemma 2.

The following example shows that $(CGLP)_k$ may admit optimal solutions that do not satisfy the condition of Theorem 2.

Example 1 *Consider the disjunctive program*

$$\max_{x_1,x_2} x_2 - 0.1x_1 \tag{32}$$
$$s.t.\ x_1 \geq 0 \tag{33}$$
$$-x_1 \geq -2 \tag{34}$$
$$x_2 \geq 0 \tag{35}$$
$$-x_2 \geq -2 \tag{36}$$
$$x_1 + 10x_2 \geq 1 \tag{37}$$
$$-x_1 + 10x_2 \geq -1 \tag{38}$$
$$-x_1 - 10x_2 \geq -1 \ \vee\ x_1 - 10x_2 \geq 1. \tag{39}$$

Observe that the two terms of disjunction (39) are the negations of the inequalites (37) and (38). The polytope of the LP relaxation is shown in Figure 1. The feasible solutions of DP lie on the line segments between $(0, 1/10)$ *and* $(1, 0)$*, and between* $(1, 0)$ *and* $(2, 1/10)$*. The optimal LP solution is* $(\hat{x}_1, \hat{x}_2) = (0, 2)$*. Then we have* $A\hat{x} - b = (0, 2, 2, 0, 19, 21)^T$*, and* $d^{1,1}\hat{x} - d_0^{1,1} = -19$*. Therefore,* $(CGLP)'_1$ *is as follows:*

$$\min 2u_2 + 2u_3 + 19u_5 + 21u_6 - 19u_0$$
$$(u-v)\begin{pmatrix} 1 & 0 & 0 \\ -1 & 0 & -2 \\ 0 & 1 & 0 \\ 0 & -1 & -2 \\ 1 & 10 & 1 \\ -1 & 10 & -1 \end{pmatrix} + u_0 \begin{pmatrix} -1 & -10 & -1 \end{pmatrix} - v_0 \begin{pmatrix} 1 & -10 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$$
$$u\mathbb{1} + v\mathbb{1} + u_0 + v_0 = 1$$
$$u, v, u_0, v_0 \geq 0$$

The optimal objective function value of $(CGLP)'_1$ *is* -4.75*, and the optimal solution is* $u_1 = v_2 = u_0 = v_0 = 1/4$*, and* $u_i = v_i = 0$ *otherwise. This yields the cut* $x_2 \leq 1/10$*, see Figure 1 (a). However, this cut cannot be derived as a disjunctive cut from any (feasible or infeasible) basis of the LP relaxation using formula (16). The reason is that the optimal basis of* $(CGLP)'_1$ *is*

$$\begin{matrix} u_1 \\ v_2 \\ u_0 \\ v_0 \end{matrix} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 2 & 1 \\ -1 & -10 & -1 & 1 \\ -1 & 10 & -1 & 1 \end{pmatrix},$$

where the first two rows correspond to the inequalities (33) and (34), respectively, and the last two rows to the two terms of the disjunction. Hence, by Theorem 1, in any basis of the LP from which $x_2 \leq 1/10$ *may be derived by formula (16), the slack of both of (33) and (34) must be nonbasic, which is impossible.*

Furthermore, there are only two bases of the small tableau from which a disjunctive cut cutting off $\hat{x}$ *can be derived. One is* $x^1 = (0, 2)$ *(the slacks of (33) and (36) are nonbasic), the other is* $x^2 = (2, 2)$ *(the slacks of (34) and (36)* *are nonbasic). The other bases (feasible and infeasible as well) are not suitable for cutting off $\hat{x}$. The corresponding cuts are*

$$19x_1 - 210x_2 \geq -21 \tag{40}$$
$$-19x_1 - 210x_2 \geq -59. \tag{41}$$

The two cuts are shown in Figure 1 (b). These cuts can of course be represented as basic feasible solution of $(CGLP)'_1$, but the corresponding objective function values are -3.99 and -3.61, respectively, so they are not optimal solutions. Clearly, (40) and (41) are dominated by $-10x_2 \geq -1$ on the LP relaxation, but they do not admit a convex combination which dominates $-10x_2 \geq -1$. Hence, $\{-10x_2 \geq -1\}$ strictly dominates the set of cuts $\{$(40), (41) $\}$ on the LP relaxation.

Theorems 1, and 2 and Example 1 imply the following:

Corollary 1 *Fix a disjunction $d^{k,1}x \geq d_0^{k,1} \;\vee\; d^{k,2}x \geq d_0^{k,2}$. The set of lift-and-project cuts dominates on the LP relaxation of (DP) the set of disjunctive cuts (16) from the bases of the simplex tableau. If the disjunction is not a split, and $(CGLP)_k$ admits (optimal) solutions that do not satisfy the condition of Theorem 2, then the dominance may be strict.*

For disjunctions of the form $x_k \leq 0 \vee x_k \geq 1$, Balas and Perregaard have proved that any lift-and-project cut is equivalent to a disjunctive cut (16) in some (possibly infeasible) basis of the small tableau. Therefore, the two sets of cuts are equivalent. It is easy to generalize their results to arbitrary split disjunctions, the details are omitted.

Finally, we provide a non-trivial characterization of the bases of $CGLP$.

Proposition 4 *The variables $(u_{M_1}, v_{M_2}, u_0, v_0)$ with $|M_1|+|M_2| = n$ constitute a feasible basis of $(CGLP)'_k$ such that both u_0 and v_0 take positive values in the corresponding basic solution if and only if there exists a nonnegative vector $w \in \mathbb{R}^{n+2}$ with $w_{n+1}, w_{n+2} > 0$ such that $wG = [0_{n+1}, 1]$ and for each i with $w_i > 0$, the $(n+1) \times (n+1)$ matrix consisting of all the rows of G but row i, and all the columns but the last one, is nonsingular, where G is the submatrix corresponding to the variables $(u_{M_1}, v_{M_2}, u_0, v_0)$, i.e.,*

$$G = \begin{pmatrix} \tilde{A}_{M_1} & \tilde{b}_{M_1} & \mathbb{1} \\ -\tilde{A}_{M_2} & -\tilde{b}_{M_2} & \mathbb{1} \\ d^{k,1} & d_0^{k,1} & 1 \\ -d^{k,2} & -d_0^{k,2} & 1 \end{pmatrix}.$$

Proof First we prove necessity. Let $w^* := (u^*_{M_1}, v^*_{M_2}, u^*_0, v^*_0)$ be the values of the basic variables $(u_{M_1}, v_{M_2}, u_0, v_0)$. We claim that if $w^*_i > 0$, then the $(n+1) \times (n+1)$ submatrix G^- of G consisting of all the rows but row i, and all the columns but the last one is nonsingular. Suppose it is not the case, and let $\tilde{w} \neq 0$ be a nonzero vector with $\tilde{w}_i = 0$ such that $\tilde{w}G = [0_{n+1}, \tilde{w}\mathbb{1}]$ (we do not require that $\tilde{w}\mathbb{1}$ be equal to 1). Since $w^*_i > 0$ and $\tilde{w}_i = 0$, we have $w^* + \lambda_1 \tilde{w} \neq \lambda_2 w^*$ for any $\lambda_1 \in \mathbb{R} \setminus \{0\}$, $\lambda_2 \in \mathbb{R}$. Hence, there exist $\lambda, \mu \in \mathbb{R}$ such that $(w^* + \lambda\tilde{w})\mathbb{1} \neq 0$, and $\mu(w^* + \lambda\tilde{w})G = [0_{n+1}, 1]$ and $\mu(w^* + \lambda\tilde{w}) \neq w^*$. This contradicts the fact that G is a basic submatrix of $(CGLP)'_k$.

Conversely, assume there exists a nonnegative vector $w \in \mathbb{R}^{n+2}$ with $w_{n+1}, w_{n+2} > 0$ such that $wG = [0_{n+1}, 1]$ and for each i with $w_i > 0$, the submatrix G^- of G consisting of all the rows of G but row i, and all the columns but the last one is nonsingular. We claim that w is a basic feasible solution of $(CGLP)'_k$, i.e., w is the unique solution of the system $wG = [0_{n+1}, 1]$. Suppose it is not the case, and let $w' \neq w$ be such that $w'G = [0_{n+1}, 1]$. We distinguish two cases:

- There is an index i with $w_i > 0$ and $w'_i \neq 0$. Then there exists $\lambda \in \mathbb{R}$ such that $w_i - \lambda w'_i = 0$. Hence, the vector $\tilde{w} := w - \lambda w'$ satisfies the following two conditions: (i) $\tilde{w}G = [0_{n+1}, 1-\lambda]$, and (ii) $\tilde{w}_i = 0$, while $w_i > 0$. This contradicts the assumption that the submatrix G^- of G consisting of all the rows but row i, and all the columns but the last one is nonsingular.

- For all row indices i with $w_i > 0$, $w'_i = 0$. This again contradicts the assumption on G and w, unless all the coordinates of w are positive, in which case $w' = 0$, which contradicts $w'\mathbb{1} = 1$.

No more cases are possible, and the statement is proved. □

4 Computations in the small tableau

The results of this section are generalizations of those of [8] for mixed 0/1 integer linear programming. The formulas become more intricate, since we allow more general disjunctions than $x_k \leq 0 \vee x_k \geq 1$.

The main idea is that we perform pivots in the small tableau in order to get a stronger cut of the form (16). To this end, we can use the procedure of Balas and Perregaard [8] with appropriate modifications. Let $\hat{s}$ be a basic feasible solution of LP corresponding to the basic variables B, and nonbasic variables N, such that at least one disjunctive constraint is violated. Suppose $k \in \Gamma$ identifies a violated disjunctive constraint, i.e., $\hat{s}^{k,1} < 0$ and $\hat{s}^{k,2} < 0$. This gives a violated disjunctive cut (16), which will be improved subsequently. For each $i \in B$ we determine the reduced cost values rc_{u_i} and rc_{v_i} in $(CGLP)'_k$. If $rc_{u_i} < 0$ or $rc_{v_i} < 0$, then we try to find a pivot column $\ell \in N$, such that after exchanging i and ℓ, the disjunctive cut in the new basis B' has a larger violation than the current cut. If such a column is found, we proceed with the new basis, otherwise we proceed with a new row with negative reduced cost, if such a row exists, otherwise the procedure stops.

We summarise the cut-generation procedure in Algorithm 1. The loop is repeated as long as a more violated cut is found than the actual one. Upon termination, the cut is computed in the space of structural variables, i.e., all surplus variables with $\pi_j \neq 0$, $j \in N$, are replaced by the corresponding inequalities of LP.

Notice that the algorithm may terminate without finding the optimal solution of $(CGLP)'_k$, since it may occur that there are rows with negative reduced cost, but no improving column is found. We emphasize that this cannot happen in the case of split disjunctions. To illustrate this, consider our example again.

Example 1 *(cont.) One may verify that in the basis for $\hat{x}^1 = (0, 2)$ there is only one row with negative reduced cost, namely, the row of the basic variable x_2 (the slack of (36) is nonbasic). But no improving pivot exists.*

Algorithm 1 Cut-generation

Require: simplex tableau of LP, feasible basis B, basic solution $\hat{s}$, disjunctive constraint $k \in \Gamma$ such that $\hat{s}^{k,1} < 0$ and $\hat{s}^{k,2} < 0$.

Ensure: disjunctive cut $\alpha x \geq \beta$ violated by $\hat{s}$.

1: Compute the disjunctive cut $\pi s_N \geq \pi_0$ with respect to N.
2: $cutviol := \pi_0 - \pi \hat{s}_N$, $maxviol := 0$.
3: **while** $cutviol > maxviol$ **do**
4: $maxviol := cutviol$.
5: Determine $i \in B$ with negative reduced cost rc_{u_i} or rc_{v_i} in $(CGLP)'_k$.
6: **if** no $i \in B$ exists with negative reduced cost rc_{u_i} or rc_{v_i} **then**
7: **goto** line 19.
8: **end if**
9: Find $\ell \in N$ such that pivoting on (i, ℓ) yields a new basis in which the disjunctive cut (16) has a violation larger than *maxviol*.
10: **if** an improving column is found **then**
11: **goto** step 15
12: **else**
13: Choose another row with negative reduced cost and **goto** step 9. If no more rows with negative reduced cost exist, **goto** step 19.
14: **end if**
15: Pivot on (i, ℓ) in the simplex tableau of LP. $B := B \setminus \{i\} \cup \{\ell\}$, $N := N \setminus \{\ell\} \cup \{i\}$.
16: Compute the disjunctive cut $\pi s_N \geq \pi_0$ with respect to N.
17: $cutviol := \pi_0 - \pi \hat{s}_N$.
18: **end while**
19: Determine $\alpha x \geq \beta$ from $\pi s_N \geq \pi_0$ by substitutions into surplus variables.

In the next two subsections we explain how to select the basic variable $i \in B$ to leave the basis and then how to choose the nonbasic variable $\ell \in N$ to enter the basis.

4.1 Computation of reduced costs

Given the set of nonbasic variables in the current simplex tableau, by Thoerem 1 there is at least one feasible basis $(u_{M_1}, v_{M_2}, u_0, v_0)$ of $(CGLP)'_k$ such that $M_1 \cup M_2 = N$. That is, for $j \in N$, let $\Delta_j = \bar{d}_j^{k,2}\bar{d}_0^{k,1} - \bar{d}_j^{k,1}\bar{d}_0^{k,2}$. To induce a feasible basic solution, M_1 has to contain all the $j \in N$ with $\Delta_j > 0$, and M_2 has to contain all the $j \in N$ with $\Delta_j < 0$. However, those $j \in N$ with $\Delta_j = 0$ can be distributed arbitrarily between M_1 and M_2. To simplify the presentation, we introduce new notation: $\bar{a}_{k_1 j} = \bar{d}_j^{k,1}$, $\bar{a}_{k_2 j} = \bar{d}_j^{k,2}$ for $j \in N \cup \{0\}$, and $\hat{s}_{k_1} := \hat{s}^{k,1}$, $\hat{s}_{k_2} := \hat{s}^{k,2}$.

Lemma 3 *Let $\hat{s}$, B, and N be the basic solution, the set of basic, and the set of nonbasic variables, respecively, in the current simplex tableau such that $\bar{a}_{k_1 0}, \bar{a}_{k_2 0} < 0$. Moreover, let $(u_{M_1}, v_{M_2}, u_0, v_0)$ be a set of basic variables of $(CGLP)'_k$. The reduced costs of the variables u_i and v_i with $i \in B$ can be computed as*

$$rc_{u_i} = \sum_{j \in M_1} \bar{a}_{ij}\hat{s}_j - \sigma(1 + \xi_i) - \frac{\bar{a}_{i0}\omega}{\theta} + \hat{s}_i \qquad (42)$$

$$rc_{v_i} = \sum_{j \in M_1} -\bar{a}_{ij}\hat{s}_j - \sigma(1 - \xi_i) + \frac{\bar{a}_{i0}\omega}{\theta} \qquad (43)$$

where

$$\xi_i = \sum_{j \in M_1} \bar{a}_{ij} - \sum_{j \in M_2} \bar{a}_{ij},$$

$$\tau_1 = \sum_{j \in M_1} \bar{a}_{k_1 j} - \sum_{j \in M_2} \bar{a}_{k_1 j},$$

$$\tau_2 = \sum_{j \in M_1} \bar{a}_{k_2 j} - \sum_{j \in M_2} \bar{a}_{k_2 j},$$

$$\theta = \bar{a}_{k_1,0}(1 + \tau_2) + \bar{a}_{k_2,0}(1 - \tau_1),$$

$$\sigma = \sum_{j \in M_1} \frac{(\bar{a}_{k_2 j}\bar{a}_{k_1 0} - \bar{a}_{k_1 j}\bar{a}_{k_2 0})\hat{s}_j}{\theta} - \frac{\bar{a}_{k_2 0}\hat{s}_{k_1}}{\theta},$$

$$\omega = (1 + \tau_2)\sum_{j \in M_1} \bar{a}_{k_1 j}\hat{s}_j + (1 - \tau_1)\sum_{j \in M_1} \bar{a}_{k_2 j}\hat{s}_j + (1 + \tau_2)\hat{s}_{k_1}.$$

Proof Our first goal is to compute a dual solution for the basis. Let G be the basis submatrix of $(CGLP)'_k$ corresponding to the variables (M_1, M_2, u_0, v_0). Observe that the objective function coefficients of the basic variables are $c^* = (\hat{s}_{M_1}, 0_{M_2}, \hat{s}_{k_1}, 0)$. Therefore, the dual solution of $(CGLP)'_k$ in the basis is $y^* = c^* G^{-1}$.

Recall the following decomposition technique for computing the inverse of a square matrix G given in block form:

$$G^{-1} = \begin{pmatrix} D & F \\ E & C \end{pmatrix}^{-1} =$$
$$\begin{pmatrix} D^{-1} + D^{-1}F(C - ED^{-1}F)^{-1}ED^{-1} & -D^{-1}F(C - ED^{-1}F)^{-1} \\ -(C - ED^{-1}F)^{-1}ED^{-1} & (C - ED^{-1}F)^{-1} \end{pmatrix}.$$

This formula can always be applied when G and submatrix D are invertible. In our case the basis is

$$G = \begin{pmatrix} I_{M_1} & & \bar{a}^T_{k_1,M_1} & -\bar{a}^T_{k_2,M_1} \\ & -I_{M_2} & \bar{a}^T_{k_1,M_2} & -\bar{a}^T_{k_2,M_2} \\ & & \bar{a}_{k_1 0} & -\bar{a}_{k_2 0} \\ \mathbb{1} & \mathbb{1} & 1 & 1 \end{pmatrix}$$

Let $D = \begin{pmatrix} I_{M_1} & \\ & -I_{M_2} \end{pmatrix}$, $C = \begin{pmatrix} \bar{a}_{k_1 0} & -\bar{a}_{k_2 0} \\ 1 & 1 \end{pmatrix}$, and E, F are the other two blocks. Then we have

$$C - ED^{-1}F = \begin{pmatrix} \bar{a}_{k_1 0} & -\bar{a}_{k_2 0} \\ 1 - \sum_{j \in M_1} \bar{a}_{k_1 j} + \sum_{j \in M_2} \bar{a}_{k_1 j} & 1 + \sum_{j \in M_1} \bar{a}_{k_2 j} - \sum_{j \in M_2} \bar{a}_{k_2 j} \end{pmatrix}$$

One may verify that

$$(C - ED^{-1}F)^{-1} = \begin{pmatrix} \frac{1 + \sum_{j \in M_1} \bar{a}_{k_2 j} - \sum_{j \in M_2} \bar{a}_{k_2 j}}{\theta} & \frac{\bar{a}_{k_2 0}}{\theta} \\ -\frac{1 - \sum_{j \in M_1} \bar{a}_{k_1 j} + \sum_{j \in M_2} \bar{a}_{k_1 j}}{\theta} & \frac{\bar{a}_{k_1 0}}{\theta} \end{pmatrix}$$

where $\theta = \sum_{j \in M_1} (\bar{a}_{k_1 0}\bar{a}_{k_2 j} - \bar{a}_{k_2 0}\bar{a}_{k_1 j}) + \sum_{j \in M_2} (\bar{a}_{k_2 0}\bar{a}_{k_1 j} - \bar{a}_{k_1 0}\bar{a}_{k_2 j}) + \bar{a}_{k_1 0} + \bar{a}_{k_2 0}$. Letting $\tau_1 = \sum_{j \in M_1} \bar{a}_{k_1 j} - \sum_{j \in M_2} \bar{a}_{k_1 j}$, and $\tau_2 = \sum_{j \in M_1} \bar{a}_{k_2 j} - \sum_{j \in M_2} \bar{a}_{k_2 j}$, we can rewrite this as

$$(C - ED^{-1}F)^{-1} = \begin{pmatrix} \frac{1+\tau_2}{\theta} & \frac{\bar{a}_{k_2 0}}{\theta} \\ -\frac{1-\tau_1}{\theta} & \frac{\bar{a}_{k_1 0}}{\theta} \end{pmatrix}, \text{ where } \theta = \bar{a}_{k_1 0}(1 + \tau_2) + \bar{a}_{k_2 0}(1 - \tau_1).$$

We compute

$$(\hat{s}_{M_1}, 0_{M_2})D^{-1}F = (\hat{s}_{M_1}, 0_{M_2})F = (\bar{a}_{k_1,M_1}\hat{s}_{M_1}, -\bar{a}_{k_2,M_1}\hat{s}_{M_1})$$

and

$$\begin{aligned} &(\hat{s}_{M_1}, 0_{M_2})D^{-1}F(C - ED^{-1}F)^{-1} \\ &= (\bar{a}_{k_1,M_1}\hat{s}_{M_1}, -\bar{a}_{k_2,M_1}\hat{s}_{M_1}) \begin{pmatrix} \frac{1+\tau_2}{\theta} & \frac{\bar{a}_{k_2 0}}{\theta} \\ -\frac{1-\tau_1}{\theta} & \frac{\bar{a}_{k_1 0}}{\theta} \end{pmatrix} \\ &= \begin{pmatrix} \frac{(1+\tau_2)\bar{a}_{k_1,M_1}\hat{s}_{M_1} + (1-\tau_1)\bar{a}_{k_2,M_1}\hat{s}_{M_1}}{\theta} & -\frac{\sum_{j \in M_1} (\bar{a}_{k_1 0}\bar{a}_{k_2 j} - \bar{a}_{k_2 0}\bar{a}_{k_1 j})\hat{s}_j}{\theta} \end{pmatrix} \end{aligned} \tag{44}$$

Therefore, we have

$$\begin{aligned} &(\hat{s}_{M_1}, 0_{M_2})(D^{-1} + D^{-1}F(C - ED^{-1}F)^{-1}ED^{-1}) \\ &= (\hat{s}_{M_1}, 0_{M_2}) - \frac{\sum_{j \in M_1} (\bar{a}_{k_1 0}\bar{a}_{k_2 j} - \bar{a}_{k_2 0}\bar{a}_{k_1 j})\hat{s}_j}{\theta} (\mathbb{1}_{M_1}, -\mathbb{1}_{M_2}) \end{aligned} \tag{45}$$

Moreover, we have

$$(-\hat{s}_{k_1}, 0)\left(C - ED^{-1}F\right)^{-1} = \begin{pmatrix} -\frac{(1+\tau_2)\hat{s}_{k_1}}{\theta} & -\frac{\bar{a}_{k_2 0}\hat{s}_{k_1}}{\theta} \end{pmatrix}$$

Therefore,

$$(-\hat{s}_{k_1}, 0)\left(C - ED^{-1}F\right)^{-1} ED^{-1} = -\frac{\bar{a}_{k_2 0}\hat{s}_{k_1}}{\theta}(\mathbb{1}_{M_1}, -\mathbb{1}_{M_2}) \tag{46}$$

By combining (44), (45), (46), we obtain

$$y^* = \left(\hat{s}_{M_1} - \sigma\mathbb{1}_{M_1}, \sigma\mathbb{1}_{M_2}, -\frac{(1+\tau_2)\bar{a}_{k_1,M1}\hat{s}_{M_1} + (1-\tau_1)\bar{a}_{k_2,M_1}\hat{s}_{M_1} + (1+\tau_2)\hat{s}_{k_1}}{\theta}, \sigma\right)$$

Now we compute the reduced costs of variables u_i and v_i for $i \in B$:

$$\begin{aligned} rc_{u_i} &= \hat{s}_i - y^*(-\bar{a}_i, -\bar{a}_{i0}, 1) = \hat{s}_i + \bar{a}_{i,M_1}\hat{s}_{M_1} - \sigma(1+\xi_i) - \frac{\bar{a}_{i0}\omega}{\theta} \\ rc_{v_i} &= -y^*(\bar{a}_i, \bar{a}_{i0}, 1) = -\bar{a}_{i,M_1}\hat{s}_{M_1} - \sigma(1-\xi_i) + \frac{\bar{a}_{i0}\omega}{\theta} \end{aligned}$$

□

Notice that Balas and Perregaard [8] used a completely different method for determining the reduced costs. The above technique simplifies in the case of split disjunction and is an easier derivation than that of Balas and Perregaard.

For computing efficiently all the reduced costs, we fix a partitioning $M_1 \cup M_2 = J$, and compute the terms ω, σ and θ, which are independent of row i. Let $M_3 = \{j \in N \mid \bar{a}_{k_1 0}\bar{a}_{k_2 j} - \bar{a}_{k_2 0}\bar{a}_{k_1 j} = 0\}$.

Proposition 5 *Let $M_1' \cup M_2'$ be any partitioning of J such that $M_1 \setminus M_3 = M_1' \setminus M_3$ and $M_2 \setminus M_3 = M_2' \setminus M_3$. Then the values of σ and θ are the same with respect to $M_1 \cup M_2$, and $M_1' \cup M_2'$.*

Proof Concerning θ:

$$\begin{aligned} \theta &= \bar{a}_{k_1,0}(1+\tau_2) + \bar{a}_{k_2,0}(1-\tau_1) \\ &= \bar{a}_{k_1 0} + \bar{a}_{k_2 0} + \sum_{j\in M_1}(\bar{a}_{k_1 0}\bar{a}_{k_2 j} - \bar{a}_{k_2 0}\bar{a}_{k_1 j}) - \sum_{j \in M_2}(\bar{a}_{k_1 0}\bar{a}_{k_2 j} - \bar{a}_{k_2 0}\bar{a}_{k_1 j}) \\ &= \bar{a}_{k_1 0} + \bar{a}_{k_2 0} + \sum_{j\in M_1\setminus M_3}(\bar{a}_{k_1 0}\bar{a}_{k_2 j} - \bar{a}_{k_2 0}\bar{a}_{k_1 j}) - \sum_{j \in M_2\setminus M_3}(\bar{a}_{k_1 0}\bar{a}_{k_2 j} - \bar{a}_{k_2 0}\bar{a}_{k_1 j}) \\ &= \bar{a}_{k_1 0} + \bar{a}_{k_2 0} + \sum_{j\in M_1'\setminus M_3}(\bar{a}_{k_1 0}\bar{a}_{k_2 j} - \bar{a}_{k_2 0}\bar{a}_{k_1 j}) - \sum_{j \in M_2'\setminus M_3}(\bar{a}_{k_1 0}\bar{a}_{k_2 j} - \bar{a}_{k_2 0}\bar{a}_{k_1 j}) \\ &= \bar{a}_{k_1 0} + \bar{a}_{k_2 0} + \sum_{j\in M_1'}(\bar{a}_{k_1 0}\bar{a}_{k_2 j} - \bar{a}_{k_2 0}\bar{a}_{k_1 j}) - \sum_{j \in M_2'}(\bar{a}_{k_1 0}\bar{a}_{k_2 j} - \bar{a}_{k_2 0}\bar{a}_{k_1 j}). \end{aligned}$$

The proof for σ is similar. □.

Unfortunately, the above statement does not hold for ω, i.e., it matters how the elements of M_3 are distributed between M_1 and M_2.

With respect to a fixed partitioning $M_1 \cup M_2 = N$, for each row i it suffices to determine $\bar{a}_{i0}$, and compute ξ_i and $\sum_{j\in M_1}\bar{a}_{ij}\hat{s}_j$ to obtain rc_{u_i} and rc_{v_i}.

4.1.1 The mixed $0-1$ programming special case

Concerning the mixed $0-1$ programming case, it can be shown that Lemma 3 reduces to Theorem 9 of [8] and Theorem 2.9 of [15]. That is, the first two terms in the definition of rc_{u_i} and rc_{v_i} are easily seen equivalent to that of the $0-1$ case. Concerning the term $\bar{a}_{i0}\omega/\theta$, we exploit that in case of split disjunctions, $\bar{a}_{k_1j} = -\bar{a}_{k_2j}$ and $\bar{a}_{k_10} + \bar{a}_{k_20} = 1$, which implies $\tau_1 = -\tau_2$, and

$$\begin{aligned}
\frac{\bar{a}_{i0}\omega}{\theta} &= \frac{\bar{a}_{i0}((1+\tau_2)\sum_{j\in M_1}\bar{a}_{k_1j}\hat{s}_j + (1-\tau_1)\sum_{j\in M_1}\bar{a}_{k_2j}\hat{s}_j + (1+\tau_2)\hat{s}_{k_1})}{\bar{a}_{k_10}(1+\tau_2) + \bar{a}_{k_20}(1-\tau_1)} \\
&= \frac{\bar{a}_{i0}((1+\tau_2)\sum_{j\in M_1}\bar{a}_{k_1j}\hat{s}_j + (1+\tau_2)\sum_{j\in M_1}-\bar{a}_{k_1j}\hat{s}_j + (1+\tau_2)\hat{s}_{k_1})}{1+\tau_2} \\
&= \bar{a}_{i0}\hat{s}_{k_1}.
\end{aligned}$$

Consequently,

$$\begin{aligned}
rc_{u_i} &= \sum_{j\in M_1}\bar{a}_{ij}\hat{s}_j - \sigma(1+\xi_i) - \bar{a}_{i0}\hat{s}_{k_1} + \hat{s}_i, \\
rc_{v_i} &= \sum_{j\in M_1}-\bar{a}_{ij}\hat{s}_j - \sigma(1-\xi_i) + \bar{a}_{i0}\hat{s}_{k_1},
\end{aligned}$$

which is equivalent to formulae (2.22) of [15].

4.2 Selection of pivot element

After selecting the pivot row based on reduced costs, an element has to be chosen to pivot on. Suppose row $i \in B$ is the pivot row and column $\ell \in N$ is chosen as pivot column. Pivoting on $\bar{a}_{i\ell}$ in the small tableau consists of adding $-\bar{a}_{r\ell}/\bar{a}_{i\ell}$ times row i to row r, for each row $r \neq i$, and multiplying row i by $1/\bar{a}_{i\ell}$. Since $i \in B$ and $\ell \in N$, this operation transforms row r as follows:

$$s_r + \gamma s_i + \sum_{j\in N}(\bar{a}_{rj} + \gamma\bar{a}_{ij})s_j = \bar{a}_{r0} + \gamma\bar{a}_{i0},$$

where $\gamma = -\bar{a}_{r\ell}/\bar{a}_{i\ell}$. In particular, after the pivot, the rows of k_1 and k_2 become

$$\begin{aligned}
s_{k_1} + \gamma_1 s_i + \sum_{j\in N}(\bar{a}_{k_1,j} + \gamma_1\bar{a}_{ij})s_j &= \bar{a}_{k_1,0} + \gamma_1\bar{a}_{i0}, \\
s_{k_2} + \gamma_2 s_i + \sum_{j\in N}(\bar{a}_{k_2,j} + \gamma_2\bar{a}_{ij})s_j &= \bar{a}_{k_2,0} + \gamma_2\bar{a}_{i0},
\end{aligned} \tag{47}$$

where $\gamma_1 = -\bar{a}_{k_1,\ell}/\bar{a}_{i\ell}$ and $\gamma_2 = -\bar{a}_{k_2,\ell}/\bar{a}_{i\ell}$. The disjunctive cut from the updated tableau rows (47) is

$$\pi_i s_i + \sum_{j\in N}\pi_j s_j \geq \pi_0, \tag{48}$$

where

$$\begin{aligned}
&\pi_0 := (\bar{a}_{k_1,0} + \gamma_1\bar{a}_{i0})(\bar{a}_{k_2,0} + \gamma_2\bar{a}_{i0})/\theta, \\
&\pi_i := \max\{\pi_i^1, \pi_i^2\}/\theta, \\
&\pi_j := \max\{\pi_j^1, \pi_j^2\}/\theta \text{ for all } j \in N, \\
&\pi_i^1 := \gamma_1(\bar{a}_{k_2,0} + \gamma_2\bar{a}_{i0}), \quad \pi_j^1 := (\bar{a}_{k_1,j} + \gamma_1\bar{a}_{ij})(\bar{a}_{k_2,0} + \gamma_2\bar{a}_{i0}), \\
&\pi_i^2 := \gamma_2(\bar{a}_{k_1,0} + \gamma_1\bar{a}_{i0}), \quad \pi_j^2 := (\bar{a}_{k_2,j} + \gamma_2\bar{a}_{ij})(\bar{a}_{k_1,0} + \gamma_1\bar{a}_{i0}).
\end{aligned}$$

The normalisation constant θ is needed to ensure that the cut is equivalent to a feasible solution of $(CGLP)'_k$. Recall from the proof of Theorem 1 that

$$\begin{aligned}\theta &= (\pi - \pi^1) + (\pi - \pi^2) + (\bar{a}_{k_2 0} + \gamma_2 \bar{a}_{i0}) + (\bar{a}_{k_1 0} + \gamma_1 \bar{a}_{i0}) \\ &= \sum_{j \in N} |\pi_j^1 - \pi_j^2| + |\pi_i^1 - \pi_i^2| + (\bar{a}_{k_2 0} + \gamma_2 \bar{a}_{i0}) + (\bar{a}_{k_1 0} + \gamma_1 \bar{a}_{i0}) \\ &= \sum_{j \in N} |\bar{a}_{k_1,j}(\bar{a}_{k_2 0} + \gamma_2 \bar{a}_{i0}) - \bar{a}_{k_2,j}(\bar{a}_{k_1 0} + \gamma_1 \bar{a}_{i0}) + \bar{a}_{ij}(\gamma_1 \bar{a}_{k_2 0} - \gamma_2 \bar{a}_{k_1 0})| \\ &\qquad + |\gamma_1 \bar{a}_{k_2 0} - \gamma_2 \bar{a}_{k_1 0}| + (\bar{a}_{k_2 0} + \gamma_2 \bar{a}_{i0}) + (\bar{a}_{k_1 0} + \gamma_1 \bar{a}_{i0}).\end{aligned}$$

Clearly, we choose that column $\ell \in N$ for which

$$\pi_i \hat{s}_i + \sum_{j \in N} \pi_j \hat{s}_j - \pi_0,$$

is minimal, and

$$\bar{a}_{k_1,0} + \gamma_1 \bar{a}_{i0} > 0 \text{ and } \bar{a}_{k_2,0} + \gamma_2 \bar{a}_{i0} > 0.$$

If $\bar{a}_{k_1,0} + \gamma_1 \bar{a}_{i0} \leq 0$ or $\bar{a}_{k_2,0} + \gamma_2 \bar{a}_{i0} \leq 0$ for all columns $\ell \in N$, then no pivot column can be chosen with respect to row i. Finally, by Theorem 1, the basis $(N \setminus \{i\}) \cup \{\ell\}$ induces a feasible solution of $(CGLP)'_k$.

Numerically the formulae of π_0, π_i and those of the π_j-s are unattractive as they contain terms with 4 numbers. We can eliminate those terms by adding $-\pi_i$ times row i of the simplex tableau to (π, π_0).

Lemma 4 *If $\pi_i^1 \geq \pi_i^2$ then adding $-\pi_i(s_i + \bar{a}_i^N s_N - \bar{a}_{i0})$ to the cut (48) yields*

$$\begin{aligned}\pi'_0 &= \bar{a}_{k_1 0}(\bar{a}_{k_2 0} + \gamma_2 \bar{a}_{i0})/\theta, \\ \pi'_i &= 0, \\ \pi'_j &= \frac{\max\{\bar{a}_{k_1,j}(\bar{a}_{k_2 0} + \gamma_2 \bar{a}_{i0}), \bar{a}_{k_2 j}(\bar{a}_{k_1 0} + \gamma_1 \bar{a}_{i0}) + \bar{a}_{ij}(\gamma_2 \bar{a}_{k_1 0} - \gamma_1 \bar{a}_{k_2 0})\}}{\theta}.\end{aligned}$$

while if $\pi_i^1 \leq \pi_i^2$, then

$$\begin{aligned}\pi'_0 &= \bar{a}_{k_2 0}(\bar{a}_{k_1 0} + \gamma_1 \bar{a}_{i0})/\theta, \\ \pi'_i &= 0, \\ \pi'_j &= \frac{\max\{\bar{a}_{k_2,j}(\bar{a}_{k_1 0} + \gamma_1 \bar{a}_{i0}), \bar{a}_{k_1 j}(\bar{a}_{k_2 0} + \gamma_2 \bar{a}_{i0}) + \bar{a}_{ij}(\gamma_1 \bar{a}_{k_2 0} - \gamma_2 \bar{a}_{k_1 0})\}}{\theta}.\end{aligned}$$

In either case, $\pi_i \hat{s}_i + \sum_{j \in N} \pi_j \hat{s}_j - \pi_0 = \pi'_i \hat{s}_i + \sum_{j \in N} \pi'_j \hat{s}_j - \pi'_0$.

Proof First suppose $\pi_i^1 \geq \pi_i^2$, i.e., $\pi_i = \pi_i^1 = \gamma_1(\bar{a}_{k_2 0} + \gamma_2 \bar{a}_{i0})$. Since

$$\theta \pi_0 = (\bar{a}_{k_2 0} + \gamma_2 \bar{a}_{i0})(\bar{a}_{k_1 0} + \gamma_1 \bar{a}_{i0}) = \bar{a}_{k_2 0}\bar{a}_{k_1 0} + \gamma_2 \bar{a}_{i0}\bar{a}_{k_1 0} + \gamma_1 \bar{a}_{i0}\bar{a}_{k_2 0} + \gamma_1 \gamma_2 \bar{a}_{i0}^2,$$

we have

$$\theta \pi_0 - \gamma_1(\bar{a}_{k_2 0} + \gamma_2 \bar{a}_{i0})\bar{a}_{i0} = \bar{a}_{k_1 0}(\bar{a}_{k_2 0} + \gamma_2 \bar{a}_{i0}).$$

To verify the formula for π_j, we compute the modified π_j^1 and π_j^2 values:

$$\theta \pi_j^1 - \theta \pi_i \bar{a}_{ij} = \theta \pi_j^1 - \gamma_1(\bar{a}_{k_2 0} + \gamma_2 \bar{a}_{i0})\bar{a}_{ij} = \bar{a}_{k_1 j}(\bar{a}_{k_2 0} + \gamma_2 \bar{a}_{i0}),$$

and

$$\begin{aligned}\theta\pi_j^2 - \theta\pi_i\bar{a}_{ij} &= \theta\pi_j^2 - \gamma_1(\bar{a}_{k_20} + \gamma_2\bar{a}_{i0})\bar{a}_{ij} \\ &= \bar{a}_{k_2j}\bar{a}_{k_10} + \gamma_2\bar{a}_{ij}\bar{a}_{k_10} + \gamma_1\bar{a}_{k_2j}\bar{a}_{i0} + \gamma_1\gamma_2\bar{a}_{ij}\bar{a}_{i0} - \gamma_1(\bar{a}_{k_20} + \gamma_2\bar{a}_{i0})\bar{a}_{ij} \\ &= \bar{a}_{k_2j}(\bar{a}_{k_10} + \gamma_1\bar{a}_{i0}) + \bar{a}_{ij}(\gamma_2\bar{a}_{k_10} - \gamma_1\bar{a}_{k_20}).\end{aligned}$$

The case of $\pi_i^1 \leq \pi_i^2$ can be verified similarly. Finally, since variable s_i is in the basis, $\hat{s}_i + \bar{a}_i^N\hat{s}_N = \bar{a}_{i0}$, and therefore, we have modified the value of (48) by 0. □

In the case of simple disjunctions, where the cuts are derived from single rows of the simplex tableau, there is an $O(n \log n)$ time procedure for finding the best pivot column. Unfortunately, such an efficient procedure is not known when the cuts are derived from pairs of rows, i.e., we can find the best pivot in $O(n^2)$ time, by mimicking each feasible pivot in the pivot row, and computing and evaluating the resulting cut.

4.3 Termination of the Lift-and-Project procedure

The Lift-and-Project procedure stops in the following cases:

1) The reduced costs of the variables u_i and v_i are nonnegative for all $i \in B$. As we will see in Section 5, this condition implies that an optimal solution of $(CGLP)'_1$ is found, see Theorem 3.

2) For each $i \in B$ with $rc_{u_i} < 0$ or $rc_{v_i} < 0$, either

   i) no pivot column can be chosen, i.e., for every $\ell \in N$, either $\bar{a}_{i\ell} = 0$, or $\bar{a}_{k_1,0} + \gamma_1\bar{a}_{i0} \leq 0$, or $\bar{a}_{k_2,0} + \gamma_2\bar{a}_{i0} \leq 0$; or

   ii) for each eligible column $\ell \in N$, pivoting on $\bar{a}_{i\ell}$ yields a Lift-and-Project cut inferior to the current one.

We mention that in case of split-disjunctions, outcome 2ii) is impossible. However, in case of general two-term disjunctions, it may indeed occur as shown below.

Example 2 *In this example we use the data of Example 1. It will be convenient to write the linear system of the LP relaxation of (32)-(39) in standard form:*

$$\begin{array}{rrrrrrcr} -x_1 & & -s_1 & & & & = & -2 \\ & -x_2 & & -s_2 & & & = & -2 \\ x_1 & +10x_2 & & & -s_3 & & = & 1 \\ -x_1 & +10x_2 & & & & -s_4 & = & -1 \end{array}$$

Notice that all the variables, structural as well as surplus, are nonnegative. The optimal basis B consists of $\{s_1, x_2, s_3, s_4\}$, the nonbasics are $N = \{x_1, s_2\}$, and the optimal simplex tableau of the LP relaxation (without reduced costs) is depicted in Table 1. The disjunctive cut from this tableau is (40), i.e., $19x1 - 210x_2 \geq -21$, and it has a violation of -3.99 (after proper normalization, see Theorem 1).

The basis of $(CGLP)'_1$ corresponding to the nonbasic variables *in the small tableau is $\{u_1, v_4, u_0, v_0\}$, and the nonbasic part of the simplex tableau for this*

$(CGLP)'_1$ var.	
u_2	v_2
u_3	v_3
u_5	v_5
u_6	v_6

var.	s_1	x_2	s_3	s_4	x_1	s_2	rhs.
s_1	1				1		2
x_2		1				1	2
s_3			1		−1	10	19
s_4				1	1	10	21

Table 1: The optimal simplex tableau of Example 1, and the nonbasic variables of $(CGLP)'_1$ corresponding the rows in the basis.

var	v_1	u_2	v_2	u_3	v_3	u_4	u_5	v_5	u_6	v_6	rhs.
u_1	−0.20	0.20	0.60	0.40	0.40	0.80	0.80			0.80	0.40
v_4	0.40	−0.40	0.80	0.20	0.20	−0.60	0.40			0.40	0.20
u_0	0.42	0.58	−0.16	0.16	0.26	0.42	−0.58	1.00		0.42	0.21
v_0	0.38	0.62	−0.24	0.24	0.14	0.38	0.38		1.00	−0.62	0.19

Table 2: The nonbasic part of the simplex tableau of $(CGLP)'_1$ in the basis $\{u_1, v_4, u_0, v_0\}$.

basis is depicted in Table 2. Only the reduced cost of v_2 is negative, i.e., $rc_{v_2} = -3.04$, and the reduced costs of all other nonbasic variable of $(CGLP)'_1$ are positive. The pivot element in the column of v_2 is unique, so v_2 should enter, and v_4 should leave the basis of $(CGLP)'_1$. The new basis consists of the variables $\{u_1, v_2, u_0, v_0\}$. The crucial observation is that the linear relaxation of (32)-(39) has no basis corresponding to the basis $\{u_1, v_2, u_0, v_0\}$ of $(CGLP)'_1$, as we have already seen in Example 1.

Now, let us examine what happens when pivoting in the small tableau. Notice that v_2, and u_2 are the variables of $(CGLP)'_1$ which correspond to the row of s_1 in the small tableau, see Table 1. The only possible pivot element in the row of s_1 is in the column of x_1. However, making this pivot would remove u_1 from the basis of $(CGLP)'_1$ (since x_1 becomes basic in the small tableau, thus both u_1 and v_1 must be nonbasic in the corresponding basis of $(CGLP)'_1$, cf. Theorem 1), and add to it v_2 (since u_2 and v_2 correspond to the row of s_1 in the small tableau). The resulting lift-and-project cut is equivalent to the disjunctive cut $-19x_1 - 210x_2 \geq -59$, and it has a violation of -3.61 (after normalization, cf. Theorem 1), which is strictly inferior to the (normalized) violation of $19x1 - 210x_2 \geq -21$, which was -3.99. To summarize, by pivoting in the small tableau only, we may definitely miss the optimal solutions of $(CGLP)'_k$, if the optimal basic solutions of $(CGLP)'_k$ cannot be represented by disjunctive cuts from some bases of the small tableau.

5 A sufficient condition for the optimality of the cut-generation method in the LP tebleau

Let B and N be the set of basic, and nonbasic variables, respectively, of the small tableau. In the corresponding basis of CGLP, the basic variables are $(u_{M_1}, v_{M_2}, u_0, v_0)$, where u_{M_1} and v_{M_2} are sets of variables with indices in M_1

and M_2, respectively, and $M_1, M_2 \subset N$ with $|M_1| + |M_2| = n$, see Theorem 1 and Proposition 4.

Since the cut-generation methods are based on the idea of the primal simplex method, a natural condition for termination is that none of the variables u_i and v_i of CGLP with $i \in B$ have a negative reduced cost. However, an important question is whether this condition is sufficient, i.e., if all the variables of CGLP corresponding to set B have non-negative reduced costs, then the current solution of CGLP is optimal. We prove next that this is indeed the case. Firstly, we prove a technical lemma.

Lemma 5 *Suppose the objective function of CGLP is* $\sum_{i\in\{0,\ldots,m+n\}}(g_i^u u_i + g_i^v v_i)$*. In any feasible basis of CGLP the reduced costs of the variables* u_i *and* v_i *with* $i = 1, \ldots, m+n$ *satisfy the condition:*

$$rc_{u_i} + rc_{v_i} = g_i^u + g_i^v - 2\sigma, \tag{49}$$

where σ *is the objective function value of the basic solution.*

Proof Let G be the basis submatrix of $CGLP$ and g' the corresponding part of the coefficients of the objective function. Then the dual solution associated with G is $\bar{y} = g'G^{-1}$. Let y_{n+2} be the dual variable corresponding to the normalization constraint of CGLP. We claim that $\bar{y}_{n+2} = \sigma$. To see this, we compute the dual objective function value of CGLP:

$$\bar{y}_{n+2} = \bar{y} \cdot e_{n+2} = g' \cdot (G^{-1} \cdot e_{n+2}) = g' \cdot \bar{z} = \sigma,$$

where e_{n+2} is the right-hand-side of the constraints of CGLP, and $\bar{z}$ is the right-hand-side multiplied by the basis inverse, i.e., the values of the basic variables of CGLP.

Now we compute rc_{u_i}, the reduced cost of u_i for arbitrary $i > 0$. The column of CGLP corresponding to u_i is $(\tilde{A}_i, b_i, 1)^T$. Therefore,

$$rc_{u_i} = g_i^u - \sum_{j=1}^{n} \bar{y}_j \tilde{A}_{ij} - \bar{y}_{n+1}\tilde{b}_i - \bar{y}_{n+2}.$$

Consequently, $\sum_{j=1}^{n} \bar{y}_j \tilde{A}_{ij} + \bar{y}_{n+1}\tilde{b}_i = g_i^u - \bar{y}_{n+2} - rc_{u_i}$.

It remains to compute rc_{v_i}. The column of CGLP corresponding to v_i is $(-\tilde{A}_i, -\tilde{b}_i, 1)^T$. Therefore,

$$rc_{v_i} = g_i^v - (\sum_{j=1}^{n} \bar{y}_j(-\tilde{A}_{ij}) - \bar{y}_{n+1}\tilde{b}_i) - \bar{y}_{n+2} = g_i^v + (g_i^u - \bar{y}_{n+2} - rc_{u_i}) - \bar{y}_{n+2}.$$

Since $\bar{y}_{n+2} = \sigma$, we have shown that $rc_{u_i} + rc_{v_i} = g_i^u + g_i^v - 2\sigma$, as claimed. □

This result has a number of consequences.

Corollary 2 *Suppose the objective function of CGLP is* $\sum_{i\in\{0,\ldots,m+n\}}(g_i^u u_i + g_i^v v_i)$*, and* $g_i^u, g_i^v \geq 0$ *for* $i = 1, \ldots, m+n$*. Moreover,* $\bar{u}_0$*,* $\bar{v}_0$*,* $\bar{u}_j$ *with* $j \in M_1$*, and* $\bar{v}_j$ *with* $j \in M_2$ *constitute a basic feasible solution of value* $\sigma < 0$*. Then for* $j \in M_1$*,* $rc_{v_j} = g_j^u + g_j^v - 2\sigma > 0$*, and for* $j \in M_2$*,* $rc_{u_j} = g_j^u + g_j^v - 2\sigma > 0$*.*

Proof Suppose $j \in M_1$. Then $rc_{u_j} = 0$ since u_j is a basic variable. Apply Lemma 5 to deduce that $rc_{v_j} = g_j^u + g_j^v - 2\sigma$. Since $g_j^u, g_j^v \geq 0$, and $\sigma < 0$ by assumption, we also have $rc_{v_j} > 0$. The argument for rc_{u_j} with $j \in M_2$ is analogous. □

Corollary 3 *Suppose u_i and v_i are both nonbasic. Then it suffices to compute one of rc_{u_i} and rc_{v_i}, the value of the other can be determined using (49).*

The main statement is as follows:

Theorem 3 *Suppose the cut-generation procedure in the small tableau terminates with $rc_{u_i} \geq 0$ and $rc_{v_i} \geq 0$ for all $i \in B$, and the corresponding basic feasible solution $w = (\bar{u}, \bar{v}, \bar{u}_0, \bar{v}_0)$ of CGLP has negative objective function value. Then w is an optimal solution of CGLP.*

Proof It suffices to verify that all assumptions of Corollary 2 are satisfied. Firstly, $g_j^u = (\tilde{A}_j x^* - \tilde{b}_j) \geq 0$ for $j = 1, \ldots, m+n$, since x^* is a feasible basic solution of LP. Moreover, $g_j^v = 0$ for $j = 1, \ldots, m+n$ by definition. We also know that σ, the objective function value of CGLP, is negative by assumption. This implies that u_0 and v_0 are basic as well. Therefore, Corollary 2 can be applied, and it implies that $rc_{v_j} > 0$ for $j \in M_1$ and $rc_{u_j} > 0$ for $j \in M_2$, where M_1 and M_2 index the basic u_j and v_j variables not including u_0 and v_0, respectively. Since $rc_{u_i}, rc_{v_i} \geq 0$ for $i \in B$, we conclude that w is a basic feasible solution with non-negative reduced costs for all of the variables. Therefore, it is an optimal solution of CGLP. □

6 Computational evaluation

6.1 The branch-and-cut procedure

The L&P cut generation method (Algorithm 1) has been embedded in a branch-and-cut algorithm for solving facial disjunctive programs with two-term disjunctions. One round of the cut-generation procedure consists of generating a disjunctive cut for each violated disjunctive constraint, but only the first 50 most violated disjunctions are considered, where the violation of a disjunction $k \in \Gamma$ is measured as $s^{k,1} \cdot s^{k,2}$. All the cuts are generated with respect to the same basic feasible solution of the linear program. After adding simultaneously all the cuts found to the linear program, it is reoptimised. After the generation of cuts, if the node cannot be fathomed by the standard rules of branch-and-cut, branching occurs. Each unfathomed node has two descendant nodes, which are obtained by adding the first and the second term, respectively, of a violated disjunction to the linear program. The disjunction $k \in \Gamma$ with largest $s^{k,1} \cdot s^{k,2}$ value is selected for branching. In another variant of the method, CGLP cuts were generated in the root node by solving $(CGLP')_k$ (system (23)) for the violated disjunctions $k \in \Gamma$.

In our experiments, in the root node of the search tree disjunctive cuts were generated in at most 3 rounds. The L&P cut generation method performed at most 50 pivots. The method always selected the first row with negative reduced cost rc_{u_i} or rc_{v_i}, and the column giving the most negative objective function value (largest violation in the row). In the search tree, disjunctive cuts were generated only in one round in the nodes of depth not greater than 4, and with one pivot only.

6.2 The test environment

The algorithm has been implemented in C++ programming language using the Xpress-MP mathematical programming package. All tests have been conducted on a PC with Pentium IV processor, 2 GHz clock speed, 512 MB RAM and Windows XP operating system.

6.3 Evaluation on LPCC instances

We have evaluated our method on test instances for Linear Programs with Complementarity Constraints (LPCCs). Given $c \in \mathbb{R}^n$, $d \in \mathbb{R}^m$, $f \in \mathbb{R}^r$, $q \in \mathbb{R}^m$, $A \in \mathbb{R}^{r\times n}$, $B \in \mathbb{R}^{r\times m}$, $M \in \mathbb{R}^{m\times m}$, and $N \in \mathbb{R}^{m\times n}$. The *linear programming with linear complementarity constraints* problem aims at finding the optimal solution $(x, y) \in \mathbb{R}^n \times \mathbb{R}^m$ of the program

$$
LPCC: \quad \begin{aligned}
&\min_{(x,y)} cx + dy \\
&\text{s.t. } Ax + By \geq f, \\
&\qquad Nx + My \geq -q, \\
&\qquad x, y \geq 0, \\
&\qquad y_k(q_k + N_k x + M_k y) = 0, \quad k = 1, \ldots, m.
\end{aligned}
$$

Clearly, LPCC is a disjunctive program with disjunctions $-y_k \geq 0 \ \vee \ -N_k x - M_k y \geq q_k$, $k = 1, \ldots, m$. We have compared our results to those of Hu et al. [12] using the test instances [16], which were obtained by a sophisticated Bender's decomposition method in which pure integer programs (IPs) were used to test whether a system of inequalities each of the form $\sum_{i\in\mathcal{I}} z_i + \sum_{j\in\mathcal{J}}(1 - z_j) \geq 1$ admits a feasible binary solution, and linear programs (LPs) were solved repeatedly, to find new extreme points, or rays of a Bender's reformulation, and also to detect infeasibility or unboundedness [12].

We tested our method on datasets with 50, and 300 complementarity constraints. In the datasets with 300 complementarity constraints, $B = 0$. We summarize our findings in Tables 3, 4, 5 and 6. In Tables 3 and 4, columns *lb* and *opt* provide the value of the LP relaxation and that of the optimal solution, respectively. The columns LPs and IPs are taken from [12] and indicate the number of linear and integer programs solved, respectively. Finally, the last four columns provide information about our method: the number of nodes explored, the number of L&P cuts generated, the total number of cut-generation rounds, and the run-time of the computation in seconds. Notice that the linear program is reoptimized once for each node and after each round of the cut-generation method. We can observe that our method solves considerably fewer linear programs than that of Hu et al. In Tables 5 and 6 we compare three variants of the method presented in this paper: pure branch-and-bound, branch-and-cut with L&P cuts, and branch-and-cut with CGLP cuts. The last row provides the average values in each table.

As can be seen, on these instances, pure branch-and-bound is the fastest method, albeit it does not rule out that a different branching strategy might give even better results when combined with cut generation. One should also note that these instances are easy as the number of nodes needed by any method is below 50 in all but one cases. In particular, on the large instances with

Table 3: Comparion of Hu et al. [12] and branch-and-cut with L&P cuts on general LPCCs with $B \neq 0$, $n = m = 50$, $r = 55$.

			Hu et al.		L&P			
#	*lb*	*opt*	LPs	IPs	nodes	cuts	rounds	time
1	28.7739	29.0501	21	2	8	19	6	0.2
2	36.1885	37.5509	229	9	25	36	6	0.34
3	33.8630	37.0022	4842	696	107	87	10	0.75
4	33.7618	34.2228	102	7	29	20	6	0.25
5	21.4187	22.2835	209	24	35	39	8	0.32
6	29.8919	30.0829	108	13	31	26	6	0.34
7	37.6712	38.0405	92	7	17	15	7	0.13
8	20.8210	22.3969	187	21	35	35	6	0.27
9	39.0227	40.3380	321	14	31	58	6	0.43
10	40.0135	41.3957	190	19	43	44	8	0.42

Table 4: Comparion of Hu et al. [12] and branch-and-cut with L&P cuts on general LPCCs with $B = 0$, $n = m = 300$, $r = 300$.

			Hu et al.		L&P			
#	*lb*	*opt*	LPs	IPs	nodes	cuts	rounds	time
1	2469.4402	2478.2256	125	1	15	26	6	7.32
2	3213.7179	3270.1844	4071	62	31	59	6	5.65
3	3639.4496	3660.5412	350	2	8	24	6	3.63
4	3127.3706	3176.4108	1249	15	25	70	6	6.91
5	2958.9144	2959.9495	5	1	2	7	1	1.45
6	2630.3286	2672.5709	4511	70	29	47	6	6.66
7	2616.985	2617.2638	0	0	2	2	1	0.69
8	2766.9542	2771.2372	26	1	7	17	6	2.37
9	2842.4483	2847.6926	319	2	7	11	4	2.75
10	3207.6865	3230.9896	1569	16	10	26	6	3.54

$n = m = r = 300$, branch-and-bound generates 15.7 nodes on average, while with L&P cuts the average is 13.6, and with CGLP cuts it is only 11.2. On the other hand, the method with L&P cuts terminates much faster than that with CGLP cuts, but the number of search tree nodes is usually a bit higher when using the L&P cut generation procedure with the above parameters.

On large instances, in about one-third of the cases L&P cuts decrease the number of nodes significantly compared to branch-and-bound, which shows that either stronger cuts are needed, or one should consider harder instances where cut-generation pays off. We also note that in the 0/1 case, the modularization technique of Balas [2] is applied to L&P cuts, and it gives excellent results, see e.g. [3],[6], but this technique is not available for the cuts emerging in LPCCs.

Table 5: Comparion of pure B&B, branch-and-cut with L&P cuts, and branch-and-cut with CGLP cuts on general LPCCs with $B \neq 0$, $n = m = 50$, $r = 55$.

	B&B		L&P			CGLP		
#	nodes	time (s)	nodes	cuts	time (s)	nodes	cuts	time (s)
1	8	0.05	8	19	0.2	8	13	0.34
2	33	0.08	25	36	0.34	25	26	0.58
3	111	0.2	107	87	0.75	91	63	1.15
4	29	0.05	29	20	0.25	35	27	0.58
5	31	0.06	35	39	0.32	19	26	0.62
6	27	0.06	31	26	0.34	31	23	0.5
7	17	0.06	17	15	0.13	13	8	0.27
8	45	0.07	35	35	0.27	27	19	0.45
9	35	0.09	31	58	0.43	31	50	0.82
10	41	0.09	43	44	0.42	41	23	0.55
avg.	37.7	0.08	36.1	37.9	0.345	32.1	27.8	0.586

Table 6: Comparion of pure B&B, branch-and-cut with L&P cuts, and branch-and-cut with CGLP cuts on general LPCCs with $B = 0$, $n = m = 300$, $r = 300$.

	B&B		L&P			CGLP		
#	nodes	time (s)	nodes	cuts	time (s)	nodes	cuts	time (s)
1	15	0.81	15	26	7.32	11	24	33.89
2	39	1.49	31	59	5.65	31	58	64.98
3	15	0.87	8	24	3.63	8	22	31.48
4	25	1.3	25	70	6.91	23	77	66.66
5	7	0.43	2	7	1.45	1	4	7.91
6	23	0.99	29	47	6.66	15	51	51.52
7	3	0.41	2	2	0.69	2	3	7.74
8	13	0.59	7	17	2.37	3	14	20.96
9	4	0.52	7	11	2.75	7	12	23.17
10	13	0.78	10	26	3.54	11	24	27.29
avg.	15.7	0.82	13.6	28.9	4.10	11.2	28.9	33.56

7 Final remarks

In this paper we have described a simple generalization of the method of Balas and Perregaard for strengthening cuts from split disjunctions. Although our generalization has some limitations, but it is still able to generate cutting planes competitive with those obtained by solving the Cut Generating Linear Program to optimality by an LP solver. However, new application areas are needed in order to really benefit from these cuts in practice.

Acknowledgements

The author is grateful to the Associate Editor and to the two anonymous referees for constructive comments that helped to improve the paper. This work has been supported by the János Bolyai Research Grant BO/00412/12/3.

References

[1] Andersen, K, Cornuéjols, G., Li, Y., Split closure and intersection cuts, Mathematical Programming, Ser A, 102 (2005) 457-493.

[2] Balas, E., Disjunctive Programming, Annals of Discrete Mathematics, 5 (1979) 3-51.

[3] Balas, E., Ceria, S., Cornuéjols, G., A lift-and-project cutting plane algorithm for mixed 0-1 programs, Mathematical Programming, 58 (1993) 295-324.

[4] Balas, E., Ceria, S., Cornuéjols, G., Mixed 0-1 programming by lift-and-project in a branch-and-cut framework, Management Science, (1996).

[5] Balas, E., Disjunctive programming: Properties of the convex hull of feasible points, Discrete Applied Mathematics, 89 (1998) 3-44.

[6] Balas, E., Projection and lifting in combinatorial optimization, In: M. Jünger, D. Naddef (eds.), Computational Combinatorial Optimization, LNCS 2241 (2001) 26-56.

[7] Balas, E., Perregaard, M., Lift-and-project for mixed $0-1$ programming: recent progress. Discrete Applied Mathematics 123 (2002) 129-154.

[8] Balas, E., Perregaard, M., A precise correspondence between lift-and-project cuts, simple disjunctive cuts, and mixed-integer Gomory cuts for 0-1 programming, Mathematical Programming, Ser B, 94 (2003) 221-245.

[9] Balas, E., Bonami, P., New variants of lift-and-project cut generation from the LP tableau: open source implementation and testing, IPCO 2007, LNCS 4513 (2007) 89-103.

[10] Balas, E., Cornuéjols, G., Kis, T., Nannicini, G., Combining Lift-and-Project and Reduce-and-Split, INFORMS J. on Computing, in press, 2012.

[11] Ceria, S., Pataki, G., Solving integer and disjunctive programs by lift and project, IPCO VI, LNCS 1412 (1998) 271-283.

[12] Hu, J., Mitchell, J.E., Pang, J-S., Bennett, K.P., Kunapuli, G., On the Global Solution of Linear Programs with Linear Complementarity Constraints, SIAM J. Optimization, 19 (2008) 445-471.

[13] Jeroslow, R.G., Cutting-planes for complementarity constraints, SIAM J. Control and Optimization, 16 (1978) 56-62.

[14] Perregaard, M., Balas, E., Generating cuts from multiple-term disjunctions, In: K. Aaardal, B. Gerards (eds.), IPCO 2001, LNCS 2081 (2001) 348-360.

[15] Perregaard, M., Generating Disjunctive Cuts for Mixed Integer Programs, PhD Dissertation, Carnegie Mellon University, Graduate School of Industrial Administration, 2003.

[16] http://www.rpi.edu/∼mitchj/generators/lpcc/.