An algorithm for the separation of two-row cuts [1]

Quentin Louveaux[2] Laurent Poirrier[2]

September 12, 2012

Abstract

We consider the question of finding deep cuts from a model with two rows of the type $P_I = \{(x,s) \in \mathbb{Z}^2 \times \mathbb{R}^n_+ : x = f + Rs\}$. To do that, we show how to reduce the complexity of setting up the polar of $\text{conv}(P_I)$ from a quadratic number of integer hull computations to a linear number of integer hull computations. Furthermore, we present an algorithm that avoids computing all integer hulls. A polynomial running time is not guaranteed but computational results show that the algorithm runs quickly in practice.

Keywords Integer Programming, Cutting Planes, Multi-row Cuts

1 Introduction

Cutting plane generation has become an important part of integer programming solvers. It allows one to automatically strengthen the linear programming relaxation of a given formulation of a mixed-integer program. Most cutting plane techniques that are implemented and computationally effective deal with inequalities that are either problem-specific or derived from one-row relaxations of the initial problem. In recent years however, a renewed interest has developed in generating cuts from several rows of a mixed-integer program. A recurrent model that has been studied in this framework consists in considering a subset of the rows of a simplex (optimal) tableau in which one relaxes the nonnegativity of the integer basic variables and the integrality of the nonbasic variables. This can be seen as a continuous relaxation of the corner polyhedron introduced by Gomory and Johnson [23, 21, 22]. Andersen, Louveaux, Weismantel and Wolsey [4] and Cornuéjols and Margot [14] studied the case of two rows and they showed that the facet-defining inequalities for the convex hull of all mixed-integer solutions of the relaxed model are intersection cuts obtained from lattice-free polyhedra in $\mathbb{R}^2$ with at most four sides. This result has been generalized to an arbitrary number of rows of the simplex tableau by Borozan and Cornuéjols in [12] where they show that the facet-defining inequalities are intersection cuts from lattice-free polyhedra in $\mathbb{R}^m$ in general. Many authors have studied variants of the model, considering for example bounds on nonbasic variables [2] or bounds on basic variables [18, 8, 20]. Some papers also considered the problem of strengthening the inequalities by considering the integrality of the nonbasic variables, the so-called lifting problem [17, 13]. This line of research has also generated some theoretical work comparing the strength of such inequalities with split cuts [5, 10, 16].

Another question that arises in this context regards how to generate and use such cutting planes computationally. Espinoza [19] performed some early experiments, generating cutting planes using three families of lattice-free polyhedra in $\mathbb{R}^m$. More recently, Dey, Lodi, Tramontani and Wolsey [15] and Basu, Bonami, Cornuéjols and Margot [9] also tackled the question, by focusing on a particular type of parametric lattice-free polyhedron in $\mathbb{R}^2$. Specifically, both make use of so-called Type-2 triangles, whose precise shape is determined by a heuristic procedure in [15], while [9] concentrates on the case

[1] This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its authors.

[2] Montefiore Institute, University of Liège

where one of the two considered rows has zero as a right-hand side. In this paper we also consider the question of how to generate these cuts computationally. However, our approach is different from [19], [9] and [15] as we do not fix a priori the shape or the type of lattice-free polyhedra used in order to generate a cut. In particular, consider a model of the type $P_I = \{(x, s) \in \mathbb{Z}^2 \times \mathbb{R}^n_+ : x = f + \sum_{j=1}^n r^j s_j\}$ with $f \in \mathbb{R}^2 \setminus \mathbb{Z}^2, r^j \in \mathbb{R}^2$. We study the separation problem i.e. given $(\hat{x}, \hat{s}) \in \mathbb{R}^2 \times \mathbb{R}^n_+$, either show that $(\hat{x}, \hat{s}) \in \text{conv}(P_I)$ or provide a valid inequality $\sum_{j=1}^n \alpha_j s_j \geq 1$ for $\text{conv}(P_I)$ that separates $(\hat{x}, \hat{s})$, or in other words such that $\sum_{j=1}^n \alpha_j \hat{s}_j < 1$. The separation problem for the 2-row model is known to be solvable in polynomial time by explicitly writing the polar of $\text{conv}(P_I)$ [3], but that approach reveals difficult to apply efficiently in practice. More specifically, the polar system of $\text{conv}(P_I)$ can be constructed by considering a set of constraints for every pair of nonbasic variables (i, j). Given (i, j), there is one constraint in the polar for every vertex of the convex hull of $\{(x, s) \in \mathbb{Z}^2 \times \mathbb{R}^2_+ : x = f + r^i s_1 + r^j s_2\}$. Constructing the polar therefore requires a quadratic number of two-dimensional integer hull computations. There are two main results in our paper. Our first result is to show that the complexity of the polar can be reduced from a quadratic to a linear (in n) number of integer hull computations in order to perform an exact separation. Our second result is to provide an algorithm that avoids computing explicitly all integer hulls and hence obtain a method that runs quickly in practice despite having no guaranteed polynomial running time.

The paper is organized as follows. Section 2 introduces the notation and the main tool to tackle the separation question, namely the polar system. We then prove the main theorem in this paper, i.e. that we can reduce the complexity of stating the polar from a quadratic number to a linear number of constraints (in n). In Section 3, we show how to further reduce the number of constraints by avoiding to compute explicitly all two-dimensional integer hulls. Finally, Section 4 presents computational results obtained with an implementation of our algorithm. In particular, we show that cut generation is fast in practice, and closes a measurable amount of gap on top of one-row intersection cuts, although much of that gap closure can be achieved with split cuts of the same rank.

2 Reducing the complexity of setting up the polar of $\text{conv}(P_I)$

We deal with the MIP problem

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \\ & x_j \in \mathbb{Z}, \ \forall j \in J. \end{array} \tag{1}$$

Consider a basis of the corresponding linear relaxation

$$x_i + \sum_{j \in N} \overline{a}_{ij} x_j = \overline{b}_i, \ \forall i \in B. \tag{2}$$

If we select two rows of (2) where the basic variables are required to be integer, relax the nonnegativity of these two basic variables, and relax the integrality of all nonbasic variables, we obtain the model discussed by Andersen et al. [4], namely the set $P_I = \{(x, s) \in \mathbb{Z}^2 \times \mathbb{R}^n_+ : x = f + \sum_{j \in N} s_j r^j\}$. Note that this set is a relaxation of the initial MIP (1), thus any valid inequality for $\text{conv}(P_I)$ is also valid for (1). For ease of presentation, we assume without loss of generality in the rest of the paper that no two vectors r^i, r^j are parallel with the same direction, i.e. $r^j \neq \mu r^i$ for all i, j and $\mu \geq 0$. We consider the question of the separation for $\text{conv}(P_I)$ using its polar, which we now introduce. To do so, we start with some useful definitions and notation. First, given a pair of indices (i, j), C_{ij} denotes the conic polyhedron with apex f and two extreme rays r^i and r^j.

Notation 1. $C_{ij} := \{x \in \mathbb{R}^2 : x = f + r^i s_i + r^j s_j, \ s_i, s_j \geq 0\}$.

Next, we introduce a notation for the set of all vertices of the integer hull of C_{ij}.

Definition 1. *We define $\mathcal{X}_{ij}$ as the set of vertices of* $\text{conv}(C_{ij} \cap \mathbb{Z}^2)$.

Once we fix i and j, every point $x \in \mathbb{R}^2$ has a unique representation as $x = f + r^i s_i + r^j s_j$. That representation is very useful to set up the polar and is defined next.

Definition 2. *Let $x, r^i, r^j \in \mathbb{R}^2$. We define $s^x_{i,j}$ and $s^x_{j,i}$ to be such that*

$$x = f + s^x_{i,j}\, r^i \; + \; s^x_{j,i}\, r^j.$$

These values exist and are unique unless $r^i = \nu r^j$, for some $\nu \in \mathbb{R}$.

Observe that $s^x_{i,j}, s^x_{j,i} \geq 0$ if $x \in C_{ij}$. The following definition is similar except that it deals with rays.

Definition 3. *Let $r^i, r^j, r^k \in \mathbb{R}^2$. We define $\lambda^j_{i,k}$ and $\lambda^j_{k,i}$ to be such that*

$$r^j = \lambda^j_{i,k}\, r^i \; + \; \lambda^j_{k,i}\, r^k.$$

These values exist and are unique unless $r^i = \nu r^k$, for some $\nu \in \mathbb{R}$.

What we refer to as the polar here is related to the 1-polar in [26] and is defined in more detail in Appendix A. Let $P \subseteq \mathbb{R}^n_+$ be a polyhedron that does not contain 0, and whose recession cone is $\mathbb{R}^n_+$. Its polar Q is the set

$$Q := \{\alpha \in \mathbb{R}^n : \alpha^T x \geq 1 \text{ is a valid inequality for } P\}. \tag{3}$$

It is a polyhedron in $\mathbb{R}^n_+$ and it has the following property: every extreme point $\bar{\alpha}$ of Q is such that $\bar{\alpha}^T x \geq 1$ is a facet-defining inequality for P, and every other facet-defining inequality for P is of the form $\bar{\beta}^T x \geq 0$ where $\bar{\beta}$ is an extreme ray of Q. Furthermore, the polar of the polar is the polyhedron itself, so a symmetric relationship holds: the facet-defining inequalities of Q are $\bar{x}^T \alpha \geq 1$ for every extreme point $\bar{x}$ of P, and $\bar{r}^T \alpha \geq 0$ where $\bar{r}$ is an extreme ray of P.

We are now ready to express the polar of $\text{conv}(P_I)$. The following statements have been proven in [4]:

1. The dimension of $\text{conv}(P_I)$ is n.
2. The extreme rays of $\text{conv}(P_I)$ are (r^j, e_j) for $j \in N$, where e_j denotes the j^{th} unit vector.
3. The vertices of $\text{conv}(P_I)$ take the form:

$$(x, s) = (f + s_j r^j + s_k r^k,\ s_j e_j + s_k e_k), \text{ with } x \in \mathcal{X}_{ij} \text{ and } s \in \mathbb{R}^n_+.$$

Note that, since P_I is defined with two equality constraints, it can be observed that the x variables are not needed in the representation. Therefore, if we project $\text{conv}(P_I)$ onto the space of the s variables, we easily obtain the full-dimensional polyhedron $\text{proj}_s(\text{conv}(P_I))$, whose facet-defining inequalities also define facets of $\text{conv}(P_I)$. Every vertex of $\text{conv}(P_I)$ can be projected onto a vertex of $\text{proj}_s(\text{conv}(P_I))$. Hence, we can formulate the polar Q of $\text{proj}_s(\text{conv}(P_I))$ as follows:

$$\begin{aligned} Q = \{\ \alpha \in \mathbb{R}^n \ \ :\ & s^T\alpha \geq 1, \quad \forall (x,s) \text{ extreme point of } \text{conv}(P_I) \\ & t^T\alpha \geq 0, \quad \forall (r,t) \text{ extreme ray of } \text{conv}(P_I)\} \end{aligned}$$

or, with our notation,

$$\begin{aligned} Q = \{\ \alpha \in \mathbb{R}^n \ \ :\ & s^x_{i,j}\alpha_i + s^x_{j,i}\alpha_j \geq 1, \quad \forall i,j,\ \forall x \in \mathcal{X}_{ij}, \\ & \alpha_i \geq 0, \quad \forall i \qquad \}. \end{aligned}$$

This allows us to consider the question of separating a point (x^*, s^*), namely either proving that $(x^*, s^*) \in \text{conv}(P_I)$, or finding $\alpha^* \in Q$ such that ${\alpha^*}^T s^* < 1$. In order to do so, we need to solve the linear optimization problem

$$\begin{aligned} \min \quad & {s^*}^T \alpha \\ \text{s.t.} \quad & \alpha \in Q. \end{aligned} \tag{4}$$

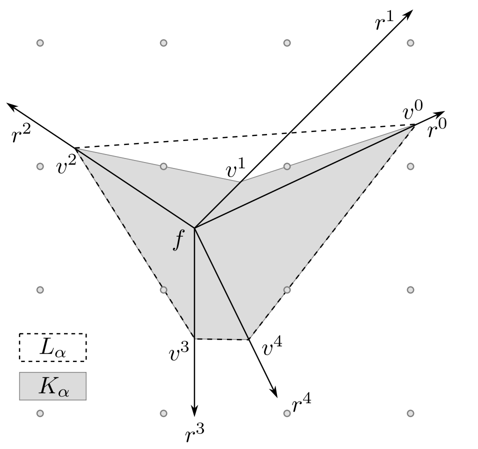

Figure 1: K_α is lattice-free while L_α is not

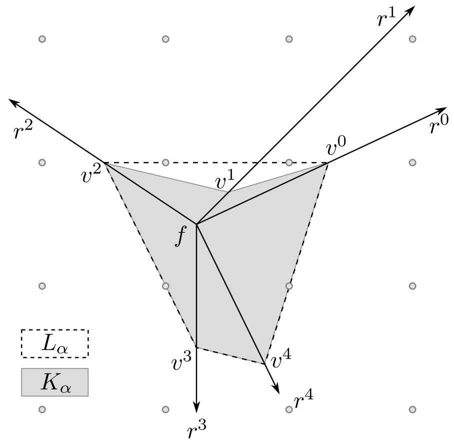

Figure 2: Both L_α and K_α are lattice-free, but K_α is not convex

Observe that (4) is bounded as long as (x^*, s^*) belongs to the linear relaxation of P_I, as then $s^* \geq 0$. If the optimal value of (4) is greater than or equal to 1, this implies that $(x^*, s^*) \in \text{conv}(P_I)$, otherwise the optimal solution α^* of (4) yields a violated valid inequality for $\text{conv}(P_I)$.

While it is possible to optimize over the polar Q, we present an alternative, more compact formulation $\overline{Q}$. Indeed, to set up Q, we need to consider every pair (r^i, r^j) of rays and compute the vertices of the integer hull of the cone C_{ij} that they define (i.e. the set $\mathcal{X}_{ij}$). Every such vertex (every point in every $\mathcal{X}_{ij}$) generates one constraint of Q. On the other hand, we construct $\overline{Q}$ by considering only pairs of *consecutive* rays (r^i, r^{i+1}) and their respective $\mathcal{X}_{i,i+1}$, plus at most n constraints linking the α coefficients for triples of consecutive rays. More precisely, we consider the set

$$\overline{Q} = \{\, \alpha \in \mathbb{R}^n_+ \ : \ s^x_{i,i+1}\alpha_i + s^x_{i+1,i}\alpha_{i+1} \geq 1, \qquad \forall i,\ \forall x \in \mathcal{X}_{i,i+1}, \tag{5}$$

$$\alpha_i \leq \lambda^i_{i-1,i+1}\alpha_{i-1} + \lambda^i_{i+1,i-1}\alpha_{i+1},\ \forall i : \ r^i \in \text{cone}(r^{i-1}, r^{i+1}) \,\}, \tag{6}$$

where the rays are indexed in counter-clockwise order, and modulo n (e.g. $r^{-1} \equiv r^{n-1}$). We observe that the set Q is described by $\sum_{i=0}^{n-1} \sum_{j=i+1}^{n-1} |\mathcal{X}_{ij}|$ constraints while $\overline{Q}$ features at most $n + \sum_{i=0}^{n-1} |\mathcal{X}_{i,i+1}|$ constraints. Theorem 1 shows that optimizing over Q can be done through optimizing over $\overline{Q}$.

Theorem 1. *Let $c \in \mathbb{R}^n$, $c > 0$, the problem $\min\{c^T\alpha \ : \ \alpha \in Q\}$ and the problem $\min\{c^T\alpha \ : \ \alpha \in \overline{Q}\}$ share the same set of optimal solutions.*

Before proving Theorem 1 we present some geometric intuition on the result. Let $P_{LP} := \{(x, s) \in \mathbb{R}^2 \times \mathbb{R}^n_+ : x = f + Rs\}$ be the linear relaxation of P_I, we define the two-dimensional lattice-free polyhedron L_α as follows.

Definition 4. *Let $\alpha \in \mathbb{R}^n_+$. The polyhedron L_α is such that its interior is the projection on the plane (x_1, x_2) of the points $(x, s) \in P_{LP}$ that violate the inequality $\alpha^T s \geq 1$:*

$$L_\alpha := \{x \in \mathbb{R}^2 : \ \textit{there exists } s \in \mathbb{R}^n_+ \text{ s.t.} (x, s) \in P_{LP} \ \textit{and } \alpha^T s \leq 1\}.$$

Definition 5. *Assume that $\alpha_i > 0$ for all i, we define $v^i := f + \frac{1}{\alpha_i} r^i$.*

To simplify the geometric discussion, we assume for the time being that $\alpha > 0$, which implies that L_α is bounded, and that the rays $\{r^i\}$ span $\mathbb{R}^2$. Observe that $L_\alpha = \text{conv}(\{v^0, \ldots, v^{n-1}\})$ (see Figure 2). The geometric intuition behind the constraints of Q is that stating $\alpha \in Q$ is equivalent to stating that L_α is lattice-free. Instead, if we want $\alpha \in \overline{Q}$, we first consider only the constraints of Q that correspond to cones formed with consecutive rays. In other words, we state that $K_\alpha := \bigcup_i \text{conv}(\{f, v^i, v^{i+1}\})$ is lattice-free (Figure 1). Note that K_α is not necessarily convex; but we can observe that if it is, then $L_\alpha = K_\alpha$.

This motivates the inclusion of the n additional constraints (6) in $\overline{Q}$, which enforce the convexity of K_α. Indeed, for K_α to be nonconvex, there must exist two consecutive triangles whose union is nonconvex, like $\text{conv}(\{f, v^0, v^1\})$ and $\text{conv}(\{f, v^1, v^2\})$ in Figure 1. The n additional constraints enforce that any v^i must be farther from f than the point in the line segment joining v^{i-1} and v^{i+1} along the half line $f + \text{cone}(r^i)$. In the presence of such constraints, we miss some valid solutions $\alpha \in Q$, i.e. those which correspond to a nonconvex K_α (Figure 2). However, in every such solution, there is one $v^i = f + r^i/\alpha_i$ in the interior of L_α, and the cut can be trivially strengthened by decreasing α_i until v^i is on the border of L_α.

We prove Theorem 1 by showing that $\overline{Q}$ is a subset of Q (Lemma 2) and that every optimal solution to $\min\{c^T\alpha : \alpha \in Q\}$, is feasible for $\overline{Q}$ (Lemma 3). First, we need the following result which shows that when (6) holds, a similar constraint also holds for non-consecutive rays contained in the same cone. In other words, it is sufficient to impose convexity constraints on *consecutive* triangles of K_α in order to obtain convexity of K_α.

Lemma 1. *If, for all j such that $r^j \in \text{cone}(r^{j-1}, r^{j+1})$,*

$$\alpha_j \le \lambda^j_{j-1,j+1}\alpha_{j-1} + \lambda^j_{j+1,j-1}\alpha_{j+1},$$

then for all i, j, k such that $r^j \in \text{cone}(r^i, r^k)$,

$$\alpha_j \le \lambda^j_{i,k}\alpha_i + \lambda^j_{k,i}\alpha_k.$$

Proof. We prove it by induction on $p := k - i \pmod n$. If $p = 0$, $p = 1$ or $p = 2$, the result is true by hypothesis. We now prove that

$$\begin{array}{llll} \text{if} & \forall i,j,l: \quad 2 \le l - i < p \pmod n & \text{and } r^j \in \text{cone}(r^i, r^l), & \alpha_j \le \lambda^j_{i,l}\alpha_i + \lambda^j_{l,i}\alpha_l \\ \text{then} & \forall i,j,l: \quad l - i = p \pmod n & \text{and } r^j \in \text{cone}(r^i, r^l), & \alpha_j \le \lambda^j_{i,l}\alpha_i + \lambda^j_{l,i}\alpha_l. \end{array}$$

Let j, k be such that $r^j, r^k \notin \{r^i, r^l\}$, $r^j \neq r^k$ and $r^j, r^k \in \text{cone}(r^i, r^l)$. Without loss of generality, we can assume that $r^j \in \text{cone}(r^i, r^k)$ and $r^k \in \text{cone}(r^j, r^l)$, i.e.

$$r^j = \lambda^j_{i,k} r^i + \lambda^j_{k,i} r^k \tag{7}$$

$$r^k = \lambda^k_{j,l} r^j + \lambda^k_{l,j} r^l \tag{8}$$

$$\lambda^j_{i,k}, \lambda^j_{k,i}, \lambda^k_{j,l}, \lambda^k_{l,j} \ge 0 \tag{9}$$

hence, using (8) in (7),

$$r^j = \lambda^j_{i,k} r^i + \lambda^j_{k,i}(\lambda^k_{j,l} r^j + \lambda^k_{l,j} r^l)$$

$$(1 - \lambda^j_{k,i}\lambda^k_{j,l}) r^j = \lambda^j_{i,k} r^i + \lambda^j_{k,i}\lambda^k_{l,j} r^l.$$

This describes r^j in terms of r^i and r^l, giving, by definition of the λ symbols (Definition 3),

$$\lambda^j_{i,l} = \frac{\lambda^j_{i,k}}{1 - \lambda^j_{k,i}\lambda^k_{j,l}}, \quad \lambda^j_{l,i} = \frac{\lambda^j_{k,i}\lambda^k_{l,j}}{1 - \lambda^j_{k,i}\lambda^k_{j,l}} \tag{10}$$

and this is well defined because $r^j \in \text{cone}(r^i, r^l)$.

Since the rays are ordered, $r^k \neq r^l$ and $r^k \in \text{cone}(r^i, r^l)$, we know that $k - i < p \pmod n$. Similarly, $l - j < p \pmod n$. Therefore, we can write, using the induction hypothesis,

$$\alpha_j \le \lambda^j_{i,k}\alpha_i + \lambda^j_{k,i}\alpha_k \tag{11}$$

$$\alpha_k \le \lambda^k_{j,l}\alpha_j + \lambda^k_{l,j}\alpha_l \tag{12}$$

hence, replacing α_k in (11) by the right-hand side of (12) and given that $\lambda^j_{k,i} \ge 0$ in (9), we obtain the new inequality

$$\alpha_j \le \lambda^j_{i,k}\alpha_i + \lambda^j_{k,i}(\lambda^k_{j,l}\alpha_j + \lambda^k_{l,j}\alpha_l)$$

which can be rewritten as

$$(1 - \lambda^j_{k,i}\lambda^k_{j,l})\alpha_j \leq \lambda^j_{i,k}\alpha_i + \lambda^j_{k,i}\lambda^k_{l,j}\alpha_l.$$

Since $\lambda^j_{i,l}, \lambda^j_{l,i} \geq 0$, given their expression in (10), we know that $(1 - \lambda^j_{k,i}\lambda^k_{j,l}) \geq 0$ and

$$\alpha_j \leq \frac{\lambda^j_{i,k}}{1 - \lambda^j_{k,i}\lambda^k_{j,l}}\alpha_i + \frac{\lambda^j_{k,i}\lambda^k_{l,j}}{1 - \lambda^j_{k,i}\lambda^k_{j,l}}\alpha_l$$

or equivalently, using again the expressions in (10),

$$\alpha_j \leq \lambda^j_{i,l}\alpha_i + \lambda^j_{l,i}\alpha_l.$$

We can prove similarly that $\alpha_k \leq \lambda^k_{i,l}\alpha_i + \lambda^k_{l,i}\alpha_l$ which concludes the induction for $l - i = p$. □

Lemma 2. *$\overline{Q}$ is a subset of Q.*

Proof. Consider $\alpha \in \mathbb{R}^n_+$ such that $\alpha \in \overline{Q}$. Some constraints of Q do not belong to the description of $\overline{Q}$. We must prove that they are satisfied. Let $x \in C_{ij}$ with $j = i + 1 \pmod n$. We consider all h, k such that $x \in C_{hk}$. Obviously, $C_{ij} \subseteq C_{hk}$, thus $r^i, r^j \in \text{cone}(r^h, r^k)$. In particular, using Lemma 1, we have

$$\alpha_i \leq \lambda^i_{h,k}\alpha_h + \lambda^i_{k,h}\alpha_k \tag{13}$$

$$\alpha_j \leq \lambda^j_{h,k}\alpha_h + \lambda^j_{k,h}\alpha_k. \tag{14}$$

Using the description of $\overline{Q}$, we also have

$$s^x_{i,j}\alpha_i + s^x_{j,i}\alpha_j \geq 1. \tag{15}$$

We now need to prove that $s^x_{h,k}\alpha_h + s^x_{k,h}\alpha_k \geq 1$.

Using Definition 2, we can express x in terms of f, r^i, r^j. And since $r^i, r^j \in \text{cone}(r^h, r^k)$, we can use Definition 3 to express them in terms of r^h, r^k:

$$\begin{aligned} x &= f + s^x_{i,j}r^i + s^x_{j,i}r^j & (16)\\ r^i &= \lambda^i_{h,k}r^h + \lambda^i_{k,h}r^k & (17)\\ r^j &= \lambda^j_{h,k}r^h + \lambda^j_{k,h}r^k & (18) \end{aligned}$$

hence, using (17)-(18) in (16),

$$x = f + (s^x_{i,j}\lambda^i_{h,k} + s^x_{j,i}\lambda^j_{h,k})\ r^h + (s^x_{i,j}\lambda^i_{k,h} + s^x_{j,i}\lambda^j_{k,h})\ r^k$$

which gives an expression of x in terms of r^h and r^k. Therefore, by Definition 2,

$$\begin{cases} s^x_{h,k} = s^x_{i,j}\lambda^i_{h,k} + s^x_{j,i}\lambda^j_{h,k} \\ s^x_{k,h} = s^x_{i,j}\lambda^i_{k,h} + s^x_{j,i}\lambda^j_{k,h} \end{cases}. \tag{19}$$

Using (13)-(14) in (15), since $s^x_{i,j}, s^x_{j,i} \geq 0$, we obtain

$$s^x_{i,j}(\lambda^i_{h,k}\alpha_h + \lambda^i_{k,h}\alpha_k) + s^x_{j,i}(\lambda^j_{h,k}\alpha_h + \lambda^j_{k,h}\alpha_k) \geq 1$$

$$(s^x_{i,j}\lambda^i_{h,k} + s^x_{j,i}\lambda^j_{h,k})\ \alpha_h + (s^x_{i,j}\lambda^i_{k,h} + s^x_{j,i}\lambda^j_{k,h})\ \alpha_k \geq 1$$

which, given (19), is equivalent to $s^x_{h,k}\alpha_h + s^x_{k,h}\alpha_k \geq 1$ □

Lemma 3. *If $c > 0$, all optimal solutions to $\min\{c^T\alpha : \alpha \in Q\}$ are feasible for $\overline{Q}$.*

Proof. Let $\alpha^* \in Q \setminus \overline{Q}$. We want to prove that α^* is not an optimal solution to $\min\{c^T\alpha : \alpha \in Q\}$. Since $\alpha^* \notin \overline{Q}$, at least one constraint of $\overline{Q}$ that is not in Q must be violated by α^*, i.e. there exists i such that

$$r^i \in \text{cone}(r^{i-1}, r^{i+1}) \text{ and } \alpha_i^* > \lambda_{i-1,i+1}^i \alpha_{i-1}^* + \lambda_{i+1,i-1}^i \alpha_{i+1}^*$$

Consider α' such that

$$\alpha'_j = \begin{cases} \alpha_j^*, & j \neq i \\ \lambda_{i-1,i+1}^i \alpha_{i-1}^* + \lambda_{i+1,i-1}^i \alpha_{i+1}^*, & j = i. \end{cases} \tag{20}$$

We claim that $\alpha' \in Q$. First, trivially, for all $j, k \neq i$ and $x \in \mathcal{X}_{jk}$,

$$s_{j,k}^x \alpha'_j + s_{k,j}^x \alpha'_k \geq 1. \tag{21}$$

Then, for all k and $x \in \mathcal{X}_{ik}$, from Definition 2 and Definition 3, we have

$$\begin{aligned} x &= f + s_{i,k}^x \; r^i + s_{k,i}^x \; r^k \\ r^i &= \lambda_{i-1,i+1}^i r^{i-1} + \lambda_{i+1,i-1}^i r^{i+1} \end{aligned}$$

with $s_{i,k}^x, s_{k,i}^x, \lambda_{i-1,i+1}^i, \lambda_{i+1,i-1}^i \geq 0$, hence

$$x = f + s_{i,k}^x \lambda_{i-1,i+1}^i r^{i-1} + s_{i,k}^x \lambda_{i+1,i-1}^i r^{i+1} + s_{k,i}^x r^k$$

is a valid representation of x. Thus $(x,\; s_{i,k}^x \lambda_{i-1,i+1}^i e_{i-1} + s_{i,k}^x \lambda_{i+1,i-1}^i e_{i+1} + s_{k,i}^x e_k) \in P_I$ and since $\alpha^* \in Q$, it must satisfy

$$\begin{aligned} s_{i,k}^x \lambda_{i-1,i+1}^i \alpha_{i-1}^* + s_{i,k}^x \lambda_{i+1,i-1}^i \alpha_{i+1}^* + s_{k,i}^x \alpha_k^* &\geq 1 \\ s_{i,k}^x (\lambda_{i-1,i+1}^i \alpha_{i-1}^* + \lambda_{i+1,i-1}^i \alpha_{i+1}^*) + s_{k,i}^x \alpha_k^* &\geq 1 \\ s_{i,k}^x \alpha'_i + s_{k,i}^x \alpha'_k &\geq 1, \end{aligned} \tag{22}$$

the third inequality being obtained because of the construction of α' in (20). Together, (21) and (22) prove that $\alpha' \in Q$. By construction, $c^T\alpha' < c^T\alpha^*$, if $c > 0$. Therefore α^* is not optimal. □

As a byproduct, the proof of Lemma 3 shows that independently of the objective function c, given $\alpha^* \in Q \setminus \overline{Q}$, there exists $\alpha' \in \overline{Q}$ which provides coefficients for a cut that strictly dominates the one based on α^*. This is the reason for requiring $c > 0$.

Proof of Theorem 1. Lemma 3 shows that all optimal solutions to $\min\{c^T\alpha : \alpha \in Q\}$ are feasible for $\overline{Q}$. Since $\overline{Q} \subseteq Q$ (Lemma 2), they correspond to the set of optimal solutions to $\min\{c^T\alpha : \alpha \in \overline{Q}\}$. □

Corollary 1. *All vertices of Q are vertices of $\overline{Q}$.*

Proof. For any vertex α^* of Q, there must exist an objective function $\bar{c}$ such that α^* is the unique optimal solution to $\min\{\bar{c}^T\alpha : \alpha \in Q\}$. For any c such that $c_i < 0$, the problem is unbounded. For any c such that $c_i = 0$, the optimal solution is not unique. Therefore, all vertices of Q can be obtained by optimizing over Q with a positive objective function, and Theorem 1 applies. □

Theorem 1 does not provide a way to tackle the case where there are zero coefficients in the objective function c, which may be important since we typically want to separate points that contain zero components. However, the following corollary holds for any $c \geq 0$.

Corollary 2. *Given $c \geq 0$, any valid inequality $\alpha^T x \geq 1$ for $\text{conv}(P_I)$ with $\alpha \in Q \setminus \overline{Q}$ is strictly dominated by a valid inequality $\hat{\alpha}^T x \geq 1$ with $\hat{\alpha} \in \overline{Q}$.*

Proof. Observe that α can be expressed as $\alpha = \hat{\alpha} + \hat{\beta}$ where $\hat{\alpha}$ is in the convex hull of the vertices of Q (thus $\hat{\alpha} \in \overline{Q}$ by Corollary 1) and $\hat{\beta}$ is in the recession cone of Q. As the recession cone of Q is in $\mathbb{R}_+^n$, it follows that $\hat{\beta} \geq 0$, hence $c^T\hat{\alpha} \leq c^T\alpha$. Since $\alpha \notin \overline{Q}$, $\hat{\beta} \neq 0$. In other words, $\hat{\alpha} \leq \alpha$ and $\hat{\alpha}_j < \alpha_j$ for some j. □

3 Separation algorithm

Optimizing over the set $\overline{Q}$ developed above requires explicit knowledge of the sets $\mathcal{X}_{i,i+1}$. More precisely, we should compute, for every $\mathrm{cone}(r^i, r^{i+1})$, the vertices of the convex hull of $\mathbb{Z}^2 \cap (f + \mathrm{cone}(r^i, r^{i+1}))$. To each of them corresponds one linear constraint of $\overline{Q}$. The number of such vertices is polynomial in the encoding length of (r^i, r^{i+1}) [24], and a polynomial-time algorithm for computing them has been presented in [3].

We adopt a fundamentally different approach that lets us avoid fully computing $\mathcal{X}_{i,i+1}$ and considering one linear constraint per point in $\mathcal{X}_{i,i+1}$. The motivation for this is to keep the linear program that we optimize over extremely small. In order to do that, we relax the expression of $\overline{Q}$, by considering constraints (5) only for x in small sets $S_{i,i+1} \subseteq C_{i,i+1} \cap \mathbb{Z}^2$ instead of in $\mathcal{X}_{i,i+1}$. Note that this is indeed a relaxation since the constraints that correspond to points in $(C_{i,i+1} \cap \mathbb{Z}^2) \setminus \mathcal{X}_{i,i+1}$ are redundant yet valid for $\overline{Q}$. We denote this relaxation by $\overline{Q}(S) \supseteq \overline{Q}$, where $S = \cup_i S_{i,i+1}$.

$$\begin{aligned}\overline{Q}(S) = \{\, \alpha \in \mathbb{R}^n_+ \;:\; & s^x_{i,i+1}\alpha_i + s^x_{i+1,i}\alpha_{i+1} \geq 1, && \forall i,\ \forall x \in S_{i,i+1} \\ & \alpha_i \leq \lambda^i_{i-1,i+1}\alpha_{i-1} + \lambda^i_{i+1,i-1}\alpha_{i+1}, && \forall i : r^i \in \mathrm{cone}(r^{i-1}, r^{i+1}) \,\}\end{aligned}$$

We then follow a classic row-generation approach summarized in Algorithm 1. First, we initialize S to a reasonable subset of $\cup_i \mathcal{X}_{i,i+1}$. We then optimize over $\overline{Q}(S)$ and find a solution α. As $\overline{Q}(S)$ is a relaxation of $\overline{Q}$, α may violate some constraints (5). If we find such a constraint, we add the corresponding point x to S, and iterate. Otherwise, if no such constraint is violated, α is valid for $\overline{Q}$ and is thus the desired optimal solution.

Initialization:	$S := S_0$
Step A:	$\bar{\alpha} := \mathrm{argmin}\{c^T\alpha \;:\; \alpha \in \overline{Q}(S)\}$
Step B:	Look for $x \in \mathbb{Z}^2$ such that $\bar{\alpha} \notin \overline{Q}(S \cup \{x\})$
	If no such x exist $\quad$ $\bar{\alpha}$ is optimal for $\min\{c^T\alpha \;:\; \alpha \in \overline{Q}\}$, terminate. Otherwise $\quad$ $S := S \cup \{x\}$, go back to Step A.

Algorithm 1: Using $\overline{Q}(S)$ to optimize over $\overline{Q}$

This scheme mainly relies on the fact that we possess an oracle that is able to find violated constraints of $\overline{Q}$, or prove that no such constraints exist. We describe such an oracle in this section. Note that the complexity of the algorithm as it is stated here is undefined, as it depends on the output of the oracle. Adopting a geometric perspective, we can restate the task of the oracle as follows: Given $\overline{Q}(S)$, $\bar{\alpha} := \mathrm{argmin}\{c^T\alpha \;:\; \alpha \in \overline{Q}(S)\}$ and the polyhedron $L_{\bar{\alpha}}$, find $x \in \mathbb{Z}^2 \cap \mathrm{interior}(L_{\bar{\alpha}})$ or prove that no such points exist. Barvinok [6] presented a polynomial-time algorithm that can solve this problem in any fixed dimension d, with the vertices of $L_{\bar{\alpha}}$ as its only input. However, we proceed otherwise, taking advantage of our specific two-dimensional setup and our knowledge of the set S. Our proposed oracle can be summarized as described in Algorithm 2. We detail the procedure in the rest of this section.

Step 1. We define $T \subseteq S$ to be the set of vertices of $\mathrm{conv}(S \cap L_{\bar{\alpha}})$. An example is shown on Figure 3. After Step A, $s^x_{i,j}\bar{\alpha}_i + s^x_{j,i}\bar{\alpha}_j \geq 1$ for all $x \in S$. Therefore, no such x lies in the interior of $L_{\bar{\alpha}}$, and points in T must be on the boundary of $L_{\bar{\alpha}}$. We want to check whether $\mathrm{conv}(T)$ is lattice-free. By triangularizing $\mathrm{conv}(T)$, the problem reduces to finding integer points on line segments and in the interior of triangles with integer vertices. Both can be solved by elementary modulo calculus. In particular, the number of integer points in the interior of a triangle $\mathrm{conv}(\{0, u, v\})$, with $u, v \in \mathbb{Z}^2$, follows directly from Pick's formula (see for example [7])

$$N_{\mathrm{interior}} = 1 + \frac{\det([u|v]) - \gcd(u_1, u_2) - \gcd(v_1, v_2) - \gcd(v_1 - u_1, v_2 - u_2)}{2} \tag{23}$$

Input:	$S \subset \mathbb{Z}^2$, $\bar{\alpha} \in \overline{Q}(S)$
Step 1:	Let T be the set of vertices of $\mathrm{conv}(S \cap L_{\bar{\alpha}})$. Check whether $\mathrm{conv}(T)$ is lattice-free.
Step 2:	Check whether there are integer points in the relative interior of the edges of $\mathrm{conv}(T)$ that are in the interior of $L_{\bar{\alpha}}$.
Step 3:	Assume $\bar{\alpha} > 0$. Use Theorem 2 to check whether $L_{\bar{\alpha}}$ is lattice-free.

Algorithm 2: Oracle for finding integer points in the interior of $L_{\bar{\alpha}}$

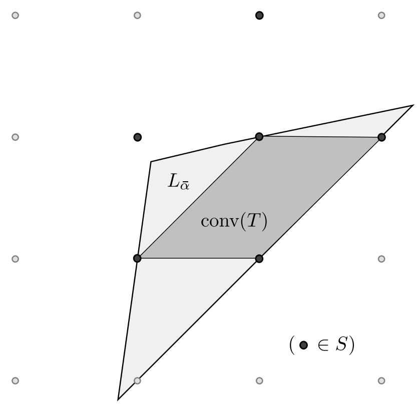

Figure 3: $T :=$ vertices of $\mathrm{conv}(S \cap L_{\bar{\alpha}})$

Then, knowing that the triangle contains integral points, we find them using the following procedure: If there are lattice points in the relative interior of two or three edges, we construct an integral point using integer combinations of these. This point is in the interior of $\text{conv}(T)$ except when there is exactly one lattice point in the relative interior of each edge, in which case we divide $\text{conv}(T)$ in the 4 sub-triangles they define, and proceed with one of these sub-triangles, as they all contain the same number of integer points in their interior. Otherwise, at least two edges contain no lattice points in their relative interior, and we make use of Lemma 4.

Lemma 4. *Let Δ be a triangle with integer vertices $\{0, u, v\}$ that has interior lattice points and such that $\gcd(u_1, u_2) = \gcd(v_1, v_2) = 1$. Δ has an interior lattice point w such that $w = \frac{1}{\det([u|v])} u + \frac{k_v}{\det([u|v])} v$ with $k_v \in \mathbb{Z}_+$.*

Proof. Any point w in $\mathbb{Z}^2$ can be expressed as $w = \beta_u u + \beta_v v$, with

$$\begin{pmatrix} \beta_u \\ \beta_v \end{pmatrix} = \begin{bmatrix} u_1 & v_1 \\ u_2 & v_2 \end{bmatrix}^{-1} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}.$$

By explicitly developing the matrix inverse, we get $\beta_u = \frac{k_u(w)}{\det([u|v])}$ and $\beta_v = \frac{k_v(w)}{\det([u|v])}$, with $k_u(w) := w_1 v_2 - w_2 v_1$ and $k_v(w) := u_1 w_2 - u_2 w_1$. Note that $k_u(w)$ and $k_v(w)$ are the components of w in the coordinate system defined by u and v, multiplied by $\det([u|v])$, and are thus integral as long as w is integer. We are looking for a point w in the interior of Δ, i.e. such that $k_u(w), k_v(w) \geq 1$ and $k_u(w) + k_v(w) \leq \det([u|v]) - 1$. The claim in this Lemma is that such a point exists even when $k_u(w)$ is fixed to 1. We now consider $k_u(w) = 1$ as a Diophantine equation with variables w_1, w_2, i.e. $w_1 v_2 - w_2 v_1 = 1$. Since $\gcd(v_2, -v_1) = 1$, there exist $\bar{w}_1, \bar{w}_2 \in \mathbb{Z}$ such that $k_u(\bar{w}) = 1$, and we could find the value of $\bar{w}$ using the Euclidian algorithm. If $1 \leq k_v(\bar{w}) \leq \det([u|v]) - 2$, then $\bar{w}$ is an interior lattice point. Otherwise, we build the integer point $w' = \bar{w} + \lambda v$, with $\lambda = -\lfloor k_v(\bar{w}) / \det([u|v]) \rfloor$. Observe that $k_u(w') = 1$ and $k_v(w') \in \{0, 1, \ldots, \det([u|v]) - 1\}$. The point w' has the desired form and is in the interior of the triangle Δ unless $k_v(w') = 0$ or $k_v(w') = \det([u|v]) - 1$. The first case, $k_v(w') = 0$, is impossible since the segment $(0, u)$ does not have lattice points in its relative interior. In the second case, we note that $w' = \frac{1}{\det([u|v])} u + \frac{\det([u|v]) - 1}{\det([u|v])} v \in \mathbb{Z}^2$. By hypothesis, there exists an integer point in the interior of Δ. Let $\widehat{w} = \frac{k_u(\widehat{w})}{\det([u|v])} u + \frac{k_v(\widehat{w})}{\det([u|v])} v$ be such a point. We know that $k_u(\widehat{w}), k_v(\widehat{w}) \geq 1$ and $k_u(\widehat{w}) + k_v(\widehat{w}) < \det([u|v]))$. Finally, we build a second point $w'' = \widehat{w} + (k_u(\widehat{w}) - 1)(v - w') = \frac{1}{\det([u|v])} u + \frac{k_u(\widehat{w}) + k_v(\widehat{w}) - 1}{\det([u|v])} v$, which proves the claim. □

Lemma 4 allows us to find an integer point $w = \frac{1}{\det([u|v])} u + \frac{k_v}{\det([u|v])} v$ in the interior of Δ by solving the Diophantine system

$$\begin{cases} u_1 + k_v v_1 = k_1 \det([u|v]) \\ u_2 + k_v v_2 = k_2 \det([u|v]) \end{cases}, \; k_v, k_1, k_2 \in \mathbb{Z}$$

for k_v, choosing the smallest positive solution. This can be done either by three applications of the Euclidean algorithm or by using the Hermite normal form of the system. In both cases, finding the smallest positive solution is easy as the set of solutions is a one-dimensional translated lattice.

Step 2. We now assume that $\text{conv}(T)$ is lattice-free, and we check that the relative interior of its edges does not contain integer points that are in the interior of $L_{\bar{\alpha}}$. Note that since $L_{\bar{\alpha}}$ is convex, it is enough to check one integer point in the relative interior of each edge of $\text{conv}(T)$.

Step 3. We now assume that no integer point was found in the interior of $L_{\bar{\alpha}}$ through Steps 1 and 2. Moreover, we assume that $\bar{\alpha}_i > 0$ for all i, hence $L_{\bar{\alpha}}$ is a polytope; we show in Section 4 how we proceed if this assumption is not true. We start by showing that $L_{\bar{\alpha}}$ is tight at three affinely independent points in S, or in other words that $\text{conv}(T) = \text{conv}(S \cap L_{\bar{\alpha}})$ is full-dimensional.

Lemma 5. *Let $\bar{\alpha}$ be a vertex of $\overline{Q}(S)$. If $\text{cone}(r^1, \ldots, r^n) = \mathbb{R}^2$ and $\bar{\alpha} > 0$, then $S \cap L_{\bar{\alpha}}$ contains three affinely independent points.*

Proof. Note that we can assume wlog that for every $x \in S$ there exists a unique i such that $x \in S_{i,i+1}$. Consider now the constraints defining the set $\overline{Q}(\mathrm{S})$.

$$
\begin{aligned}
&s^x_{i,i+1}\alpha_i + s^x_{i+1,i}\alpha_{i+1} \geq 1, && \forall i,\ \forall x \in S_{i,i+1} && (24)\\
&\alpha_i \leq \lambda^i_{i-1,i+1}\alpha_{i-1} + \lambda^i_{i+1,i-1}\alpha_{i+1}, && \forall i : r^i \in \operatorname{cone}(r^{i-1}, r^{i+1}) && (25)\\
&\alpha_i \geq 0 && \forall i && (26)
\end{aligned}
$$

There are $|S|$ constraints of type (24), n of type (25), and n of type (26), of which a subset of n linearly independent overall must be tight at $\bar{\alpha}$. Because $\bar{\alpha} > 0$, none of the nonnegativity constraints are tight for $\bar{\alpha}$. Now observe that v^i is a vertex of $L_{\bar{\alpha}}$ only if it is not on $\operatorname{aff}(v^{i-1}, v^{i+1})$. In other words, only if $\bar{\alpha}_i \neq \lambda^i_{i-1,i+1}\bar{\alpha}_{i-1} + \lambda^i_{i+1,i-1}\bar{\alpha}_{i+1}$, i.e. the associated constraint (25) is not tight.

Moreover, if $\operatorname{cone}(r^1, \ldots, r^n) = \mathbb{R}^2$, then $f \in \operatorname{interior}(L_{\bar{\alpha}})$. Therefore, $L_{\bar{\alpha}}$ is full-dimensional, and has at least three vertices. This implies that at most $n-3$ of the constraints (25) are tight for $\bar{\alpha}$. Equivalently, we have at least three tight constraints of type (24) for $\bar{\alpha}$. If the corresponding three integer points are affinely independent, the result follows.

Suppose now that they are on a line. More specifically let W be a collinear set of such tight points, i.e. $W \subseteq S \cap \operatorname{boundary}(L_{\bar{\alpha}})$ such that $|W| \geq 3$ and $\dim(\operatorname{aff}(W)) = 1$. Since $L_{\bar{\alpha}}$ is convex, all points in W must belong to a single facet $\operatorname{conv}(v^i, v^j)$ of $L_{\bar{\alpha}}$. Now let us define K as the index set of the rays inside the corresponding cone, i.e. $K := \{k : r^k \in \operatorname{cone}(r^i, r^j)\}$. Observe that there are at most $|K|$ linearly independent tight constraints including the variables with indices in K only. Thus, there must be at least $n - |K|$ linearly independent tight constraints including at least one of the $n - |K|$ remaining variables. For at least one of these remaining variables, the associated ray supports a vertex of $L_{\bar{\alpha}}$, i.e. there exists $h \notin K$ such that v^h is a vertex of $L_{\bar{\alpha}}$. Therefore, at least one of the $n - |K|$ remaining constraints (25) is not tight for $\bar{\alpha}$. It follows that there is at least one additional tight constraint (24), which does not correspond to a point in W. □

As suggested by Lemma 5, we ensure that we have three affinely independent points on the boundary of $L_{\bar{\alpha}}$ by adding artificial rays to P_I, as needed, in order to have $\mathbb{R}^2$-spanning rays. By using a zero objective function cost for variables associated to artificial rays, we do not modify the separation problem. Observe that since $\operatorname{conv}(T)$ is a lattice-free polyhedron in $\mathbb{R}^2$ with integer vertices, we know that it has at most four vertices [25], thus $|T|$ may only be three or four, i.e. $\operatorname{conv}(T)$ is a triangle or a qualidrateral.

Definition 6 summarizes the assumptions we can make in Step 3: we call $(T, L_{\bar{\alpha}})$ *checkable* if we found integer points in the interior of $L_{\bar{\alpha}}$ neither in Step 1 nor in Step 2.

Definition 6. *We call a couple* $(T, L_{\bar{\alpha}})$ checkable *if*

(a) $\operatorname{conv}(T)$ *and* $L_{\bar{\alpha}}$ *are full-dimensional convex polytopes in* $\mathbb{R}^2$*,*

(b) the vertices of $\operatorname{conv}(T)$ *are integral and belong to the boundary of* $L_{\bar{\alpha}}$*,*

(c) $\operatorname{conv}(T)$ *is lattice-free,*

(d) the integer points in the relative interior of the edges of $\operatorname{conv}(T)$ *do not belong to the interior of* $L_{\bar{\alpha}}$*.*

We showed previously that it is easy to verify whether $(T, L_{\bar{\alpha}})$ is checkable. In the remainder of this section, we show that it is computationally cheap to check whether $L_{\bar{\alpha}}$ is lattice-free when $(T, L_{\bar{\alpha}})$ is checkable. Lemma 6, 7, 8 and 9 cover the four possible cases.

Lemma 6. *Let* $(T, L_{\bar{\alpha}})$ *be checkable and* $\operatorname{conv}(T)$ *be a lattice-free triangle with exactly one integer point in the relative interior of each edge. Then* $L_{\bar{\alpha}}$ *is lattice-free.*

Proof. $\operatorname{conv}(T)$ is a maximal lattice-free body. Therefore, $L_{\bar{\alpha}} \supseteq \operatorname{conv}(T)$ is lattice-free if and only if $L_{\bar{\alpha}} = \operatorname{conv}(T)$. Since the integer points on the edges of $\operatorname{conv}(T)$ are not in the interior of $L_{\bar{\alpha}}$, they are on its boundary, so $L_{\bar{\alpha}} = \operatorname{conv}(T)$. □

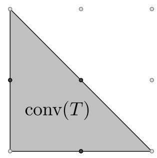

Figure 4: One lattice point in the relative interior of each edge of conv(T) (Lemma 6)

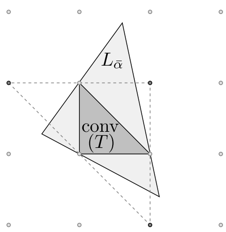

Figure 5: conv(T) is a unimodular triangle (Lemma 7)

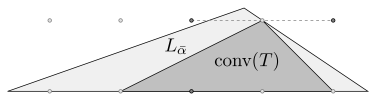

Figure 6: One or more lattice points in the relative interior of one edge of conv(T) (Lemma 8)

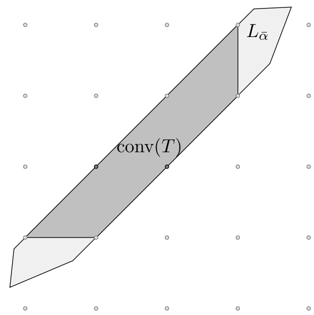

Figure 7: conv(T) is a quadrilateral (Lemma 9)

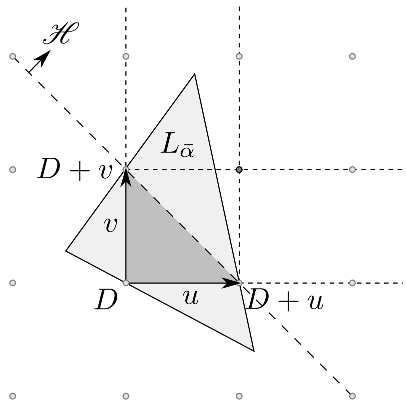

Figure 8: The half-plane $\mathcal{H}$ in the proof of Lemma 7

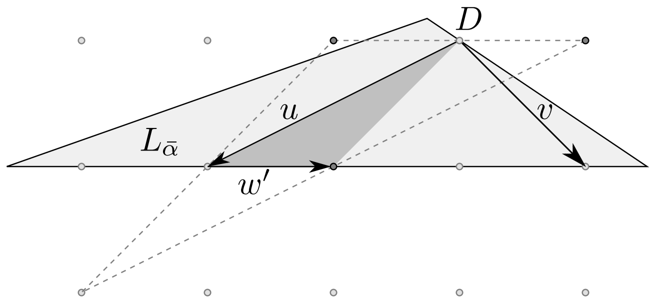

Figure 9: w' in Lemma 8

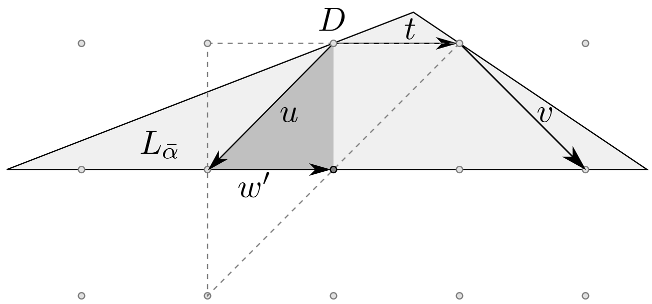

Figure 10: Unimodular triangle decomposition in Lemma 9

Lemma 7. *Let* $(T, L_{\bar{\alpha}})$ *be checkable and* $\operatorname{conv}(T)$ *be a triangle with vertices* D, $D+u$, $D+v$ *such that* $\det([u|v]) = 1$. *Then* $L_{\bar{\alpha}}$ *is lattice-free if and only if* $L_{\bar{\alpha}}$ *contains neither* $D+u+v$ *nor* $D+u-v$ *nor* $D+v-u$ *in its interior.*

Proof. Let $\mathcal{H}$ be the half-plane delimited by the line $(D+u, D+v)$ not containing D (see Figure 8). We first consider the vertices of $L_{\bar{\alpha}}$ that lie in $\mathcal{H}$. Observe that by convexity, they must belong to $D + \operatorname{cone}(u, v)$, otherwise $D+u$ or $D+v$ would belong to the interior of $L_{\bar{\alpha}}$. Since (u, v) is an integral basis of $\mathbb{Z}^2$, there is no integer point in the interior of $\operatorname{conv}(\{D, D+v\}) + \operatorname{cone}(\{u\})$ or $\operatorname{conv}(\{D, D+u\}) + \operatorname{cone}(\{v\})$. Hence, if the vertices of $L_{\bar{\alpha}}$ all lie in $\operatorname{conv}(\{D, D+v\}) + \operatorname{cone}(\{u\})$ or all lie in $\operatorname{conv}(\{D, D+u\}) + \operatorname{cone}(\{v\})$, then $L_{\bar{\alpha}} \cap \mathcal{H}$ is lattice-free. Otherwise, $D+u+v$ is in the interior of $L_{\bar{\alpha}}$. By symmetry for the two other half-planes, the result follows. $\square$

Lemma 8. *Let* $(T, L_{\bar{\alpha}})$ *be checkable and* $\operatorname{conv}(T)$ *be a lattice-free triangle with vertices* D, $D+u$, $D+v$ *such that* $\gcd(u_1, u_2) = \gcd(v_1, v_2) = 1$ *and* $\gcd(w_1, w_2) \neq 1$, *with* $w = v - u$. *Then* $L_{\bar{\alpha}}$ *is lattice-free if and only if* $L_{\bar{\alpha}}$ *contains neither* $D + w'$ *nor* $D - w'$ *in its interior, with* $w' = \frac{w}{\gcd(w_1, w_2)}$.

Proof. By convexity, the point $D + u + w'$ belongs to $L_{\bar{\alpha}}$, but since it is in the relative interior of an edge of $\operatorname{conv}(T)$, it is not in the interior of $L_{\bar{\alpha}}$. Thus, $D + u + w'$ is on the boundary of $L_{\bar{\alpha}}$ and we consider the triangle $\operatorname{conv}(\{D, D+u, D+u+w'\})$. By (23), $\det([-u, w']) = 1$ and Lemma 7 applies. Therefore, in that case, $L_{\bar{\alpha}}$ is lattice-free if and only if $D + w'$, $D - w'$ and $D + 2u + w'$ do not lie in its interior. Note that since the line $(D+u, D+v)$ is a facet of $L_{\bar{\alpha}}$, $D + 2u + w'$ can not be in $L_{\bar{\alpha}}$. $\square$

Lemma 9. *Let* $(T, L_{\bar{\alpha}})$ *be checkable and* $\operatorname{conv}(T)$ *be a lattice-free quadrilateral. Then* $L_{\bar{\alpha}}$ *is lattice-free.*

Proof. As for Lemma 8, we can decompose $\text{conv}(T)$ in unimodular triangles (Figure 10) and apply Lemma 7 on one of them. The vertices of this unimodular triangle can be vertices of $\text{conv}(T)$ or integer points in the relative interior of the edges of $\text{conv}(T)$, belonging in both cases to the boundary of $L_{\bar{\alpha}}$. One can easily see that the integer points to be verified in Lemma 7 are either on the edges of $\text{conv}(T)$ or trivially not in the interior of $L_{\bar{\alpha}}$, by convexity. □

Theorem 2. *Let $(T, L_{\bar{\alpha}})$ be checkable and* $\text{conv}(T)$ *be a 2-dimensional convex polytope of wich D, $D+u$ and $D+v$ are three vertices. Then $L_{\bar{\alpha}}$ is lattice-free if and only if it contains neither $D+u'+v'$ nor $D+u'-v'$ nor $D+v'-u'$ in its interior, with $u' = \frac{u}{\gcd(u_1,u_2)}$ and $v' = \frac{v}{\gcd(v_1,v_2)}$.*

Proof. This follows from Lemma 6, 7, 8 and 9. □

The results in this section show that, in Step 3, it is enough to check at most three integer points against the interior of $L_{\bar{\alpha}}$ to verify that $L_{\bar{\alpha}}$ is lattice-free.

4 Computations

In this section, we present an implementation of the two-row cut separator described previously and computational results obtained with it. First, we need to address an issue that was not covered before.

Handling of the case $\alpha_i = 0$. So far, we have only considered the case where $L_{\bar{\alpha}}$ is a polytope, i.e. $\bar{\alpha}_i > 0$ for all i, while the constraints of $\overline{Q}(S)$ only ensure $\bar{\alpha}_i \geq 0$. Note that if $L_{\bar{\alpha}}$ is lattice-free and unbounded, then it is necessarily a split set. If only one coefficient $\bar{\alpha}_h$ is zero, then we can check that $L_{\bar{\alpha}}$ is inscribed in a split set parallel to r^h. If two such coefficients $\bar{\alpha}_j$ and $\bar{\alpha}_k$ are zero, then $L_{\bar{\alpha}}$ is not lattice-free and we can easily construct an integer point $f + \mu_j r^j + \mu_k r^k$ in its interior. However, both these operations involve computing a rational representation of the rays r^i and the point f, which are usually only available in floating-point form. Besides numerical difficulties, this could yield points with large coefficients, and that are very "far" from f. Recall that when we find points in $\mathbb{Z}^2 \cap \text{interior}(L_{\bar{\alpha}})$, we add them to the set S, whose elements correspond to the constraints of $\overline{Q}$ that are considered in its relaxation $\overline{Q}(S)$. In that context, large coefficients cause numerical instability, and points that are "far" from f typically correspond to weak (i.e. dominated) constraints of $\overline{Q}$.

We tackle this issue in a different way, by imposing positive lower bounds on α, thus ensuring that we obtain a bounded polyhedron $L_{\bar{\alpha}}$. This can yield problems in case some of these bounds end up tight in $\bar{\alpha}$, as then the result of Lemma 5 is not guaranteed to hold, i.e. $\text{conv}(T)$ may not be full-dimensional. If it is full-dimensional nevertheless, then Theorem 2 holds and we can still use the oracle described in Algorithm 2. Otherwise, as mentioned earlier, we could fall back on verifying that $L_{\bar{\alpha}}$ is lattice-free with the polynomial-time algorithm of Barvinok [6]. But for that case, we instead implement the naive enumeration that runs in $w(L_{\bar{\alpha}})$ iterations, where

$$w(L_{\bar{\alpha}}) = \max\{x_1 \ : \ x \in L_{\bar{\alpha}}\} - \min\{x_1 \ : \ x \in L_{\bar{\alpha}}\}.$$

This is not polynomial in the encoding length of $L_{\bar{\alpha}}$, and potentially much more costly than Algorithm 2, which is basically $O(1)$ for fixed values of $|S|$ and n. In practice however, it is possible to bound the number of iterations by suitably choosing the value of the lower bounds on α, for instance $\alpha_i \geq \frac{|r^i|}{K}$ for some $K > 0$. Since the vertices of $L_{\bar{\alpha}}$ are of the form $v^i = f + \frac{1}{\bar{\alpha}_i} r^i$, this ensures that $L_{\bar{\alpha}}$ is contained in a disc of radius K centered at f, implying $w(L_{\bar{\alpha}}) \leq 2K$. In our implementation, $K = 500$. Note also that if we fix the value of K, the complexity of Algorithm 1 becomes polynomial, as it performs at most $O(K^2)$ iterations.

Remark that $L_{\bar{\alpha}}$ being lattice-free is enough for $\bar{\alpha}^T s \geq 1$ to be a valid inequality for P_I, but if some components of $\bar{\alpha}$ are at their positive lower bound, then $\bar{\alpha}^T s \geq 1$ is not guaranteed to define a *facet* of conv(P_I), as the lower bounds on α are not actual constraints of the polar of conv(P_I). For this reason, whenever we generate a valid inequality having a coefficient α_j at lower bound, we consider instead the intersection cut corresponding to a split set of direction r^j, i.e. the lattice-free body $L_\alpha = \{x \in \mathbb{R}^2 : \lfloor p^T f \rfloor \leq p^T x \leq \lceil p^T f \rceil\}$ where $p \in \mathbb{Z}^2$ is an integral vector orthogonal to r^j. This is the only facet-defining inequality for P_I having $\alpha_j = 0$, and is hence a solution to $\min\{{s^*}^T \alpha \ : \ \alpha \in Q\}$ in this case, if K is sufficiently large. The computation can fail since $p^T f$ may be integral, which is made more likely by the fact that we must convert (approximately) a floating-point representation of r^j into a rational in order to compute p. If it fails for every such α_j at lower bound, then we discard the current cut, so that we return only facet-defining inequalities.

Computational experiment. Algorithm 3 summarizes the computational experiment performed in order to measure the practical speed of our method. Given a mixed-integer problem, the algorithm

```
input: a mixed-integer problem P, its linear relaxation P_LP

for r = 1 to RANK_MAX                                        (outer loop)
        Optimize over P_LP. Let x* be the optimal solution.
        Compute the optimal simplex tableau.
        Build up to MODELS_MAX two-row models.

        for each row of the simplex tableau,
                generate the corresponding one-row intersection cut.
        end for
        Add the one-row cuts separating x* to P_LP.

        do                                                   (inner loop)
                Optimize over P_LP. Let x* be the optimal solution.

                for each two-row model,
                        generate a cut, trying to separate x*.
                end for
                Add the cuts separating x* to P_LP.

        while at least one cut was added.
end for
```

Algorithm 3: Computational experiment

starts by optimizing over its linear relaxation. In the outer loop, we extract, from the simplex tableau associated to the current optimal solution x^*, the two-row models to be used for cut generation. In the inner loop, we separate one inequality with each model and add the cuts that separate x^* to the linear relaxation, over which we then reoptimize, yielding a new solution x^*. The inner loop terminates when no more separating inequality is found. At that point, the next iteration of the outer loop will build different models based on a new simplex tableau. Observe that at a given iteration of the outer loop, all the generated inequalities are at most of rank r.

In order to compare two-row inequalities with their single-row counterpart, we compute the one-row intersection cut associated to each row of the simplex tableau, at every outer loop iteration, before we start separating two-row inequalities. These cuts are intersection cuts on a one-row relaxation of the original problem, keeping the integrality constraint only for the corresponding basic variable (i.e. *non-lifted* intersection cuts). Note that while much easier to compute, they are a subset of those we can obtain with our two-row separator.

Although more sophisticated options exist (see e.g. [19, 15, 10, 9]), our method for building the two-row models is essentially heuristic. We arbitrarily restrict ourselves to reading rows from optimal simplex tableaux. Our intent is to build models whose constraints have similar supports, while covering all

CPU:	Intel Core i7-990X at 3.47GHz, 6 cores, 12 threads
RAM:	DDR3-1333 SDRAM (24Gb)
Compiler:	GCC 4.6.3 20120306 (Red Hat 4.6.3-2)
Environment:	GNU/Linux (Fedora 15), kernel 2.6.43.8-1.fc15.x86_64
Cut generation:	Implemented in C++, single threaded
LP solver:	IBM CPLEX 12.4 (C library API), 64 bits, single threaded

Table 1: Conditions of the experiments

relevant rows. Intuitively, this can be motivated by observing that any intersection cut separated from a model whose two rows have disjoint support is equivalent to a linear combination of two intersection cuts from the corresponding one-row models. In practice, we select up to `MODELS_MAX` models meeting the following requirements:

1. Each of the two rows is a *suitable* simplex tableau row, i.e.
   a. its basic variable is integer-constrained,
   b. its density, i.e. the ratio of the number of nonzero coefficients in the row over the number of columns, does not exceed `ROW_DENSITY_MAX`.
2. Each of the two rows is used in at most (`ROW_USE_MAX` $-$ 1) other selected models.
3. At least one of the two rows has a fractional right-hand side.
4. Among the models that are not selected, none has a higher score. The score of a two-row model is computed as $(c - d)$ where c is the number of colums having nonzero coefficients in both rows, and d is the number of columns a having nonzero coefficient in exactly one row.

Results analysis. Tables 2 and 3 present the results of our experiment. The testbed is composed of problems from the MIPLIB 3 [11] and MIPLIB 2003 [1] libraries. We report results on all the instances except for three having no integrality gap (`dsbmip`, `enigma`, `disctom`), four whose optimal solution is unknown (`dano3mip`, `liu`, `momentum3`, `t1717`), and five for which the experiment runs out of memory (`ds`, `momentum2`, `stp3d`, `mzzv42z`, `rd-rplusc-21`). The general conditions of our experiments are detailed in Table 1. The columns of Tables 2 and 3 are composed of two parts. The first one (*one-row only*) serves as a comparison point using only one-row intersection cuts, i.e. split cuts from a simple disjunction on a basic variable (`MODELS_MAX = 0`, `ROW_USE_MAX = 0`). The column *cuts* indicates the number of separating one-row intersection cuts and *%gc* is the percentage of integrality gap closed as a result. We compute gap closures as

$$\%gc = 100\,\frac{z_{\text{LP+cuts}} - z_{\text{LP}}}{z_{\text{MIP}} - z_{\text{LP}}}$$

where z_{MIP} is the optimal objective function value of the original problem, z_{LP} the one of its LP relaxation, and $z_{\text{LP+cuts}}$ the one of its LP relaxation with cuts added. The second part (*one-row + two-row*) corresponds to Algorithm 3 with `MODELS_MAX = 5000` and `ROW_USE_MAX = 4`, i.e. each row of the simplex tableau is used to build at most 4 different two-row models. Note that we do not consider rows with more than 40% nonzero components (`ROW_DENSITY_MAX = 0.4`). In both cases, we limit ourselves to rank-5 inequalities (`RANK_MAX = 5`), and we discard cuts whose dynamism (i.e. the quotient of the largest and the smallest nonzero coefficient, in absolute value) exceeds 10^6, as they are likely to cause numerical difficulties. Moreover, we consider that a cut $\alpha^T x \geq 1$ "separates" a point x^* only if its violation at x^* is at least 10^{-6} i.e. $1 - \alpha^T x^* \geq 10^{-6}$. The column *one-row cuts* indicates the number of separating one-row intersection cuts generated as part of Algorithm 3. In the subcategory *two-row*, *models* indicates the overall number of times the two-row separation procedure is called, *time* shows the total time spent within the algorithm, in seconds, and *cuts* indicates the number of two-row cuts that succeed at separating the corresponding x^*. The set S in Algorithm 1 is initialized with the four points $(\lfloor f_1 \rfloor, \lfloor f_2 \rfloor)$, $(\lfloor f_1 \rfloor, \lceil f_2 \rceil)$, $(\lceil f_1 \rceil, \lfloor f_2 \rfloor)$ and $(\lceil f_1 \rceil, \lceil f_2 \rceil)$, and *+total* denotes the total number times a point was added to a set S, across all separations. Finally, *%gc* shows the percentage of gap closed by adding all the separating cuts.

instance	one-row only		one-row + two-row					
			one-row	two-row				
	cuts	%gc	cuts	models	+total	time	cuts	%gc
10teams	699	0.00	699	286	267	0.179	0	0.00
air03	36	100.00	36	232	111	0.316	0	100.00
air04	1299	9.49	1299	0	0	0.000	0	9.49
air05	1051	6.32	1051	0	0	0.000	0	6.32
arki001	163	27.28	161	14592	72606	32.034	238	32.23
bell3a	71	69.56	71	3008	16994	3.614	430	68.25
bell5	115	26.23	90	1656	8678	1.403	25	23.08
blend2	46	21.61	66	12453	5988	4.357	203	26.73
cap6000	67	54.19	67	0	0	0.000	0	54.19
danoint	100	0.43	95	4657	68866	55.138	121	0.61
dcmulti	278	58.32	206	7052	16350	6.574	370	65.64
egout	79	69.81	47	1867	3781	0.908	244	93.37
fast0507	1662	3.10	1662	453	220	0.759	0	3.10
fiber	253	17.04	195	30092	84777	37.087	401	18.81
fixnet6	93	18.66	78	27487	18291	14.743	535	53.54
flugpl	43	14.22	38	540	1428	0.353	133	20.37
gen	211	61.19	224	7957	36457	10.054	366	63.66
gesa2	290	47.25	270	22181	34055	11.352	687	70.54
gesa2_o	398	47.13	380	52715	71444	29.022	960	67.23
gesa3	324	49.72	208	12012	65236	18.045	668	74.46
gesa3_o	421	67.36	358	39177	80491	27.407	1049	74.62
gt2	79	97.54	78	817	2807	0.485	20	99.00
harp2	130	11.85	133	6612	26458	6.808	264	18.53
khb05250	53	95.57	38	588	258	0.192	43	90.67
l152lav	326	15.20	326	0	0	0.000	0	15.20
lseu	80	38.00	93	722	3144	0.540	96	36.89
markshare1	29	0.00	29	0	0	0.000	0	0.00
markshare2	34	0.00	34	0	0	0.000	0	0.00
mas74	74	4.38	74	0	0	0.000	0	4.38
mas76	77	3.06	77	0	0	0.000	0	3.06
misc03	275	4.56	293	3330	1792	1.129	100	17.36
misc06	37	63.18	52	5383	2796	1.949	140	86.35
misc07	392	0.72	352	6980	13931	5.004	24	0.72
mitre	5631	83.93	5496	125000	826666	210.567	3141	84.45
mkc	725	39.40	788	130000	36106	51.893	676	26.31
mod008	33	11.22	33	0	0	0.000	0	11.22
mod010	258	57.73	256	140	24	0.085	2	58.84
mod011	22	6.87	21	6178	5629	2.869	52	12.41
modglob	50	28.72	42	2392	13963	2.754	122	48.41
noswot	163	0.00	119	3536	5176	1.112	167	0.00
nw04	76	17.95	76	0	0	0.000	0	17.95
p0033	34	12.77	57	1046	10709	1.681	76	57.01
p0201	325	25.93	383	1502	4121	1.131	116	45.17
p0282	182	16.03	247	21296	135084	34.065	100	13.54
p0548	300	50.83	367	17816	57142	15.693	261	66.53
p2756	264	0.89	374	39743	48256	18.420	201	42.12
pk1	68	0.00	68	0	0	0.000	0	0.00
pp08a	204	77.53	158	4048	7886	3.168	217	90.16
pp08acuts	99	47.14	148	5336	23651	8.664	238	60.23
qiu	116	3.05	92	1804	38900	30.450	88	4.64
qnet1	298	22.99	288	13592	18518	8.649	45	28.01
qnet1_o	152	47.84	125	12859	24792	7.785	121	51.54
rentacar	9	0.00	9	368	1254	5.036	0	0.00
rgn	72	0.00	72	385	110	0.123	47	0.00
rout	195	7.81	194	456	414	0.205	7	9.30
set1ch	464	80.71	304	16179	20079	6.433	543	94.14
seymour	22038	14.75	22167	45000	338910	652.530	12	15.01
stein27	452	0.00	437	3248	8342	2.595	18	0.00
stein45	1069	0.00	1089	30119	119287	33.615	268	0.00
swath	278	0.60	283	4320	9450	6.401	342	2.53
vpm1	87	27.29	115	10169	2079	2.192	130	51.69
vpm2	143	38.79	159	15165	18774	7.750	322	53.56
average	695.032	29.415	691.081	12492.677	38912.065	22.344	232.726	36.180

Table 2: Time and gap closed on MIPLIB 3

instance	one-row only		one-row + two-row					
			one-row	two-row				
	cuts	%gc	cuts	models	+total	time	cuts	%gc
a1c1s1	278	36.35	219	11214	51555	12.391	529	43.86
aflow30a	154	19.99	170	9388	2211	3.455	92	21.73
aflow40b	169	11.52	179	52312	6659	22.780	112	14.30
atlanta-ip	9050	8.75	12993	120000	2936415	5566.283	446	8.75
glass4	108	0.00	178	3052	299	0.751	885	0.00
manna81	1980	100.00	1980	10000	503	3.316	0	100.00
momentum1	10043	61.03	15421	105000	5586289	3987.552	2471	57.56
msc98-ip	21799	53.94	25529	90000	823555	2052.142	308	53.95
mzzv11	18354	20.06	17285	90000	2976550	10139.824	709	20.16
net12	4180	9.25	4399	90928	845309	554.549	204	11.54
nsrand-ipx	1071	25.51	1292	44768	35315	39.955	363	29.59
opt1217	130	0.53	126	88	20	0.040	2	0.53
protfold	5651	18.24	5759	127651	145047	169.947	2006	13.65
roll3000	3158	67.18	2897	105000	957758	479.864	630	60.07
sp97ar	2585	10.79	2673	34004	11648	27.467	49	11.61
timtab1	555	27.20	353	12162	37555	18.771	676	47.84
timtab2	669	24.96	577	23686	155427	76.760	968	34.21
tr12-30	442	68.53	393	25800	302668	99.013	362	92.22
average	4465.3	31.32	5134.6	53058.5	826376.8	1291.9	600.7	34.53

Table 3: Time and gap closed on MIPLIB 2003 (instances not included in MIPLIB 3)

The primary objective of our experiment is to assess whether our separator is fast in practice. In particular, since Algorithm 1 does not have a proven complexity bound, we need to evaluate how many iterations it performs in a practical setting. Over the course of the experiment conducted to generate Table 2 and Table 3, only 10.4 points are added to S on average per call to the separator, in addition to the four initial points, i.e. Algorithm 1 performs 11.4 iterations on average. On average, the computation takes 20.5ms per call to the separator (1.8ms when considering MIPLIB 3 only). A cut is successfully computed in 99.87% of the cases, failure happening when no facet-defining inequality could be generated as explained above, and 1.39% of the generated cuts do separate the corresponding point x^*. Note that for being considered a separating cut, a valid inequality must also satisfy the condition on coefficient dynamism described previously. On the other hand, 86.61% of the generated one-row intersection cuts are separating (in the same *one-row + two-row* experiment), with separation taking approximately 0.14ms on average in our implementation. Recall however that two-row cuts are separated *after* one-row cuts, hence every separating two-row cut is, in the separation sense, "stronger" than all the one-row cuts generated before. Note finally that overall in our experiment, only 16.98% of the time is taken by the two-row separator, the rest being spent optimizing over the LP relaxation, computing the optimal LP tableau, and selecting pairs of rows.

The secondary objective of this experiment is to evaluate the usefulness of two-row cuts in a separation scheme. In terms of average gap closed (35.81%, compared to 29.85% with one-row intersection cuts only), the addition of two-row inequalities does seem to slightly strengthen the original formulation, without however providing a compelling argument to justify their computational cost. But it should be noted that on some instances (e.g. misc03, p0201, p0548, p2756, pp08a, qnet1, tr12-30), two-row cuts provide a significant improvement in the LP bound, without the addition of a disproportionate number of cuts. On a lot of other instances, the significant improvement brought by two-row cuts could be attributed to their sheer number. Remark also that in some cases, the amount of gap closed by *one-row + two-row* is smaller than with *one-row only*. This can happen since, as we do not limit ourselves to rank-1 cuts, the bases (and hence the tableaux) used in the various experiments can differ, starting from the second outer loop iteration.

Figure 11 illustrates the evolution of the average gap closed from one (outer loop) iteration to the next, over both instance sets. The bars labeled *one-row only* and *one-row + two-row* correspond to the gap closed after each iteration in their respective experiment. The *one-row (+ two-row)* bar corresponds to the gap closed at each iteration of the *one-row + two-row* experiment before the addition of two-row inequalities, i.e. at line 12 of Algorithm 3. Observe that while the difference between the two experiments is noticeable, one-row cuts still seem to close most of the gap in the *one-row + two-row* experiment. This might indicate that the main advantage provided by the two-row inequalities arises from obtaining useful relaxations from simplex bases that are not reached adding one-row cuts only.

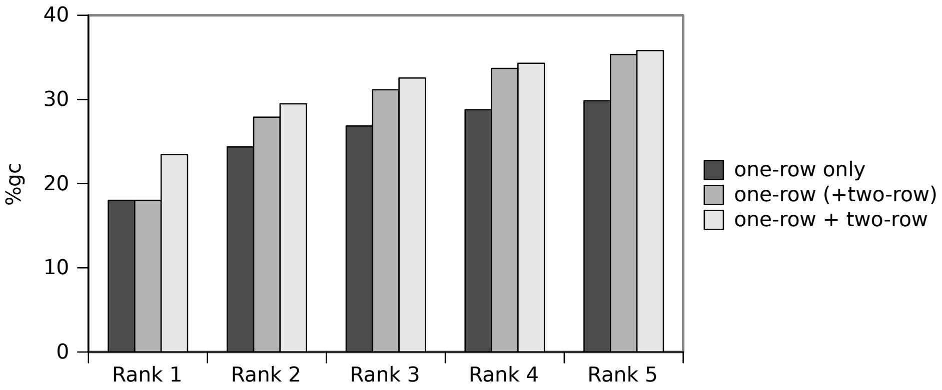

Figure 11: Gap closed, for different values of r

In Table 4 and Table 5, we survey the lattice-free sets that we obtained throughout the experiment, and classify them according to the taxonomy described in [17, 16]. We count only lattice-free sets corresponding to separating two-row cuts. One can observe that no Type-1 triangle is generated, which is worth noting as only cuts corresponding to Type-1 triangles have infinite split rank [4, 16]. This could be caused by the fact that the set P_I must take a very specific form in order for us to be able to build a Type-1 triangle with each of its vertices on $f + \text{cone}(r^i)$ for some ray r^i.

Split sets play a special role in the context of intersection cuts. In particular, any two-row intersection cut from a split set can be obtained, at a fraction of the computational cost, as a one-row cut, by combining the two initial rows. Such a cut is thus a rank-1 split cut (when constructing models from the initial LP formulation). In order to evaluate the importance of these cuts, we re-ran the experiment, discarding all two-row cuts except the ones arising from split sets. Therefore, this new experiment uses only split cuts of rank at most 5. The results are shown in Table 6 and Table 7, under the column "*+ two-row splits*". It appears that intersection cuts on split sets alone can close most of the gap closed by all two-row cuts together, going from 29.85% (*one-row only*) to 34.40% (*+ two-row splits*), instead of 35.81% (*+ two-row all*). More importantly, they do so despite the addition of fewer cuts, 135 in average (*+ two-row splits*), instead of 316 (*+ two-row all*).

5 Conclusion

In this work, we developed a compact formulation of the polar of the two-row model studied by Andersen et al. [4], built an algorithm to separate two-row cuts using this formulation, and showed computationally that it is fast in practice. As a result, we are able to generate separating two-row intersection cuts without fixing the underlying lattice-free set within a few milliseconds of computing time. We do not answer however the question of the practical usefulness of these cuts, as our experiments show mixed results in this regard. More precisely, for the instances tested, two-row intersection cuts close significantly more gap than one-row intersection cuts, but we can achieve almost as much when restricting to two-row cuts from split sets, which are split cuts of the same rank. In that context, the main direction for further research is to evaluate computationally the impact of various strengthenings of the two-row relaxation, obtained by reintroducing some of the original constraints that were dropped in it.

Acknowledgements. *We would like to thank two anonymous referees for constructive input regarding the overall presentation and our computational experiments.*

instance	cuts	T^1	T^2	T^3	Q^1	Q^2	split
10teams	0	0	0	0	0	0	0
air03	0	0	0	0	0	0	0
air04	0	0	0	0	0	0	0
air05	0	0	0	0	0	0	0
arki001	238	0	55	27	116	1	39
bell3a	430	0	100	16	211	0	103
bell5	25	0	5	0	9	0	11
blend2	203	0	61	24	101	3	14
cap6000	0	0	0	0	0	0	0
danoint	121	0	5	8	80	6	22
dcmulti	370	0	19	22	255	17	57
egout	244	0	41	12	71	6	114
fast0507	0	0	0	0	0	0	0
fiber	401	0	107	22	194	4	74
fixnet6	535	0	33	14	145	9	334
flugpl	133	0	33	5	84	3	8
gen	366	0	48	20	219	11	68
gesa2	687	0	125	24	483	14	41
gesa2_o	960	0	267	46	510	11	126
gesa3	668	0	104	22	445	17	80
gesa3_o	1049	0	89	36	840	7	77
gt2	20	0	1	0	10	0	9
harp2	264	0	65	18	156	2	23
khb05250	43	0	6	13	17	0	7
l152lav	0	0	0	0	0	0	0
lseu	96	0	47	5	35	3	6
markshare1	0	0	0	0	0	0	0
markshare2	0	0	0	0	0	0	0
mas74	0	0	0	0	0	0	0
mas76	0	0	0	0	0	0	0
misc03	100	0	9	0	47	0	44
misc06	140	0	24	9	86	0	21
misc07	24	0	3	0	5	0	16
mitre	3141	0	1135	77	1822	21	86
mkc	676	0	89	23	474	19	71
mod008	0	0	0	0	0	0	0
mod010	2	0	0	0	0	0	2
mod011	52	0	18	3	31	0	0
modglob	122	0	14	8	50	4	46
noswot	167	0	25	3	129	0	10
nw04	0	0	0	0	0	0	0
p0033	76	0	25	5	36	0	10
p0201	116	0	6	2	11	5	92
p0282	100	0	5	5	68	3	19
p0548	261	0	79	5	114	0	63
p2756	201	0	31	12	18	1	139
pk1	0	0	0	0	0	0	0
pp08a	217	0	29	37	84	10	57
pp08acuts	238	0	43	13	142	16	24
qiu	88	0	0	3	85	0	0
qnet1	45	0	6	0	29	1	9
qnet1_o	121	0	21	4	81	4	11
rentacar	0	0	0	0	0	0	0
rgn	47	0	0	7	3	1	36
rout	7	0	1	0	6	0	0
set1ch	543	0	137	125	205	10	66
seymour	12	0	3	0	3	0	6
stein27	18	0	11	0	4	1	2
stein45	268	0	27	5	182	23	31
swath	342	0	35	25	159	6	117
vpm1	130	0	38	9	37	7	39
vpm2	322	0	45	14	179	17	67
average	232.726	0.000	49.516	11.742	130.177	4.242	37.048

Table 4: Lattice-free body types (MIPLIB 3)

instance	cuts	T^1	T^2	T^3	Q^1	Q^2	split
a1c1s1	529	0	20	60	92	3	354
aflow30a	92	0	17	4	55	4	12
aflow40b	112	0	27	8	69	1	7
atlanta-ip	446	0	47	5	252	46	96
glass4	885	0	4	1	1	0	879
manna81	0	0	0	0	0	0	0
momentum1	2471	0	15	1	1	0	2454
msc98-ip	308	0	11	1	250	23	23
mzzv42z	345	0	16	2	225	27	75
net12	204	0	12	0	48	14	130
nsrand-ipx	363	0	53	17	163	2	128
opt1217	2	0	0	0	0	0	2
protfold	2819	0	97	112	2186	241	183
rd-rplusc-21	327	0	49	25	82	3	168
roll3000	630	0	61	14	386	0	169
sp97ar	49	0	15	1	10	0	23
timtab1	676	0	38	73	113	9	443
timtab2	968	0	52	92	226	18	580
tr12-30	362	0	28	248	52	1	33
average	609.895	0.000	29.579	34.947	221.632	20.632	303.105

Table 5: Lattice-free body types (MIPLIB 2003, instances not included in MIPLIB 3)

A Polarity

In the context of optimization, the term *polar* is most commonly used to denote a set describing all the valid inequalities of a polyhedron. For the remainder of this section, we denote by P the polyhedron whose polar we are interested in. In all generality, the polar of $P \in \mathbb{R}^n$ could be defined as

$$Q := \{(\alpha, \alpha_0) \in \mathbb{R}^n \times \mathbb{R} : \alpha x \geq \alpha_0 \text{ for all } x \in P\}.$$

Since we are interested in valid inequalities for $\text{conv}(P_I)$, which features particular properties, we derive here a specific polar for a family of polyhedra that includes $\text{conv}(P_I)$. For the sake of conciseness, we call them *radial* polyhedra. This polar is tightly related to the 1-polar in Nemhauser and Wolsey [26], the latter applying to full-dimensional polytopes.

Definition 7. *We call a polyhedron P* radial *if*

(a) P is not empty,

(b) P does not contain a line,

(c) P does not containt the origin 0, and

(d) for every $x \in P$, $\mu x \in P$ for all $\mu \geq 1$.

Remark that the condition (d) could alternatively be written $P = P + \text{cone}(P)$, or $P \subseteq \text{recc}(P)$, where $\text{recc}(P)$ denotes de recession cone of P.

We showed earlier that the dimension of $\text{conv}(P_I)$ is n, as is the dimension of its projection on the space of the s variables. The valid inequalities for $\text{conv}(P_I)$ thus coincide with the valid inequalities for that projection $\text{proj}_s(\text{conv}(P_I))$. Note that $0 \notin \text{proj}_s(\text{conv}(P_I))$, $\text{proj}_s(\text{conv}(P_I)) \subseteq \mathbb{R}^n_+$, and the recession cone of $\text{proj}_s(\text{conv}(P_I))$ is $\mathbb{R}^n_+$, so $\text{proj}_s(\text{conv}(P_I))$ is radial.

Through normalization of the right-hand side, valid inequalities can be divided in three classes: $\alpha^T x \geq 1$, $\beta^T x \geq 0$ and $\gamma^T x \leq 1$. Proposition 1 lets us dismiss the latter class for radial polyhedra.

Proposition 1. *Let P be a radial polyhedron. Every facet-defining inequality of P is of the form $\alpha^T x \geq 1$, $\alpha \in \mathbb{R}^n$ or $\beta^T x \geq 0$, $\beta \in \mathbb{R}^n$.*

Proof. Consider a facet-defining inequality of type $\gamma^T x \leq 1$. Either $\gamma^T x \leq 0$ for all $x \in P$, in which case $\gamma^T x \leq 1$ does not describe a proper face of P, or there exists $\bar{x} \in P$ such that $\gamma^T \bar{x} > 0$. Then, $\mu \bar{x} \in P$

instance	one-row only		+ two-row splits		+ two-row all	
			two-row		two-row	
	cuts	%gc	cuts	%gc	cuts	%gc
10teams	699	0.00	0	0.00	0	0.00
air03	36	100.00	0	100.00	0	100.00
air04	1299	9.49	0	9.49	0	9.49
air05	1051	6.32	0	6.32	0	6.32
arki001	163	27.28	65	27.97	238	32.23
bell3a	71	69.56	48	67.39	430	68.25
bell5	115	26.23	3	22.38	25	23.08
blend2	46	21.61	6	22.61	203	26.73
cap6000	67	54.19	0	54.19	0	54.19
danoint	100	0.43	28	0.57	121	0.61
dcmulti	278	58.32	24	66.62	370	65.64
egout	79	69.81	70	80.19	244	93.37
fast0507	1662	3.10	0	3.10	0	3.10
fiber	253	17.04	86	23.15	401	18.81
fixnet6	93	18.66	361	53.30	535	53.54
flugpl	43	14.22	1	14.19	133	20.37
gen	211	61.19	54	63.95	366	63.66
gesa2	290	47.25	50	57.52	687	70.54
gesa2_o	398	47.13	69	59.11	960	67.23
gesa3	324	49.72	92	73.18	668	74.46
gesa3_o	421	67.36	126	68.59	1049	74.62
gt2	79	97.54	5	97.54	20	99.00
harp2	130	11.85	10	12.43	264	18.53
khb05250	53	95.57	13	94.58	43	90.67
l152lav	326	15.20	0	15.20	0	15.20
lseu	80	38.00	3	35.33	96	36.89
markshare1	29	0.00	0	0.00	0	0.00
markshare2	34	0.00	0	0.00	0	0.00
mas74	74	4.38	0	4.38	0	4.38
mas76	77	3.06	0	3.06	0	3.06
misc03	275	4.56	43	17.38	100	17.36
misc06	37	63.18	9	72.54	140	86.35
misc07	392	0.72	16	0.72	24	0.72
mitre	5631	83.93	100	84.74	3141	84.45
mkc	725	39.40	133	35.29	676	26.31
mod008	33	11.22	0	11.22	0	11.22
mod010	258	57.73	2	58.84	2	58.84
mod011	22	6.87	12	6.99	52	12.41
modglob	50	28.72	70	44.22	122	48.41
noswot	163	0.00	15	0.00	167	0.00
nw04	76	17.95	0	17.95	0	17.95
p0033	34	12.77	7	56.76	76	57.01
p0201	325	25.93	92	44.47	116	45.17
p0282	182	16.03	15	11.40	100	13.54
p0548	300	50.83	61	61.98	261	66.53
p2756	264	0.89	136	47.64	201	42.12
pk1	68	0.00	0	0.00	0	0.00
pp08a	204	77.53	70	89.29	217	90.16
pp08acuts	99	47.14	32	59.93	238	60.23
qiu	116	3.05	41	8.35	88	4.64
qnet1	298	22.99	11	31.73	45	28.01
qnet1_o	152	47.84	20	47.05	121	51.54
rentacar	9	0.00	0	0.00	0	0.00
rgn	72	0.00	40	2.93	47	0.00
rout	195	7.81	0	7.81	7	9.30
set1ch	464	80.71	144	90.92	543	94.14
seymour	22038	14.75	15	13.27	12	15.01
stein27	452	0.00	19	0.00	18	0.00
stein45	1069	0.00	31	0.00	268	0.00
swath	278	0.60	120	1.48	342	2.53
vpm1	87	27.29	34	43.34	130	51.69
vpm2	143	38.79	58	52.48	322	53.56
average	695.032	29.415	39.677	34.791	232.726	36.180

Table 6: Two-row cuts on split sets (MIPLIB 3)

instance	one-row only		+ two-row splits		+ two-row all	
	cuts	%gc	two-row cuts	%gc	two-row cuts	%gc
a1c1s1	278	36.35	337	40.97	529	43.86
aflow30a	154	19.99	13	17.15	92	21.73
aflow40b	169	11.52	3	11.67	112	14.30
atlanta-ip	9050	8.75	53	8.74	446	8.75
glass4	108	0.00	879	0.00	885	0.00
manna81	1980	100.00	0	100.00	0	100.00
momentum1	10043	61.03	4716	61.15	2471	57.56
msc98-ip	21799	53.94	25	53.94	308	53.95
mzzv11	18354	20.06	430	21.65	709	20.16
net12	4180	9.25	179	10.08	204	11.54
nsrand-ipx	1071	25.51	166	28.54	363	29.59
opt1217	130	0.53	2	0.53	2	0.53
protfold	5651	18.24	310	16.03	2006	13.65
roll3000	3158	67.18	159	57.23	630	60.07
sp97ar	2585	10.79	45	12.93	49	11.61
timtab1	555	27.20	363	47.26	676	47.84
timtab2	669	24.96	638	35.61	968	34.21
tr12-30	442	68.53	61	71.80	362	92.22
average	4465.333	31.324	465.500	33.071	600.667	34.532

Table 7: Two-row cuts on split sets (MIPLIB 2003, instances not included in MIPLIB 3)

for all $\mu \geq 1$. In particular, choosing $\mu = \frac{2}{\gamma^T \bar{x}}$, we obtain $\gamma^T(\mu\bar{x}) = 2$, hence $\gamma^T x \leq 1$ is not a valid inequality for P. □

Furthermore, we can write a variant of the separating hyperplane theorem for radial polyhedra that involves only valid inequalities of the first class.

Proposition 2. *Given a radial polyhedron P and a point $y \notin P$, there exist a valid inequality $\alpha^T x \geq 1$ for P such that $\alpha^T y < 1$.*

Proof. By the separating hyperplane theorem, there exist a valid inequality for P that separates y. (a). If the inequality is of the form $\alpha^T x \geq 1$, then the claim is proven. (b). Assume that the inequality is of the form $\beta^T x \geq 0$. As it separates y, we know that $\beta^T y < 0$. Since $0 \notin P$, by the separating hyperplane theorem, there also exist a valid inequality separating 0. That second inequality can not be of the form $\bar{\beta}^T x \geq 0$ or $\bar{\gamma}^T x \leq 1$ as it would then not separate 0. Let $\bar{\alpha}^T x \geq 1$ be that valid inequality for P. If $\bar{\alpha}^T y < 1$ then it separates y and the claim is proven. Otherwise, $\bar{\alpha}^T y \geq 1$. We now linearly combine the two valid inequalities with the positive coefficients $\frac{\bar{\alpha}^T y}{-\beta^T y}$ and 1, yielding a third valid inequality $\tilde{\alpha}^T x \geq 1$ with $\tilde{\alpha} = \frac{\bar{\alpha}^T y}{-\beta^T y}\beta + \bar{\alpha}$. That inequality separates y since $\frac{\bar{\alpha}^T y}{-\beta^T y}\beta^T y + \bar{\alpha}^T y = 0 < 1$. (c). Assume that the inequality is of the form $\gamma^T x \leq 1$. As we have shown earlier, there does not exist $\bar{x} \in P$ such that $\gamma^T \bar{x} > 0$. Indeed, we would then have $\mu\bar{x} \in P$ for all $\mu \geq 1$. In particular, choosing $\mu = \frac{2}{\gamma^T \bar{x}}$, we would obtain $\gamma^T(\mu\bar{x}) = 2$, showing that $\gamma^T x \leq 1$ is not a valid inequality for P. Hence we can strengthen the inequality by writing $\gamma^T x \leq 0$, or equivalently $\beta^T x \geq 0$ where $\beta = -\gamma$. Using (b), we obtain a valid inequality of the desired form. □

We are now ready to write the definition of the polar of a radial polyhedron. Although we arbitrarily restrict ourselves to valid inequalities of the form $\alpha^T x \geq 1$, we will show at the end of this section why this choice preserves the generality of the definition.

Definition 8. *Let P be a radial polyhedron. The* polar *Q of P is the set of all $\alpha \in \mathbb{R}^n$ such that $\alpha^T x \geq 1$ is a valid inequality for P:*

$$Q = \left\{\alpha \in \mathbb{R}^n : \ \alpha^T x \geq 1, \ \textit{for all } x \in P\right\}$$

We next present a description of Q in terms of vertices and extreme rays of P. This is especially handy as this is the type of description that we have of $\mathrm{conv}(P_I)$.

Proposition 3 ([26] Proposition 5.1)**.** *Q is described by*

$$Q = \{ \ \alpha \in \mathbb{R}^n : \alpha^T x^k \geq 1 \textit{ for all } x^k \textit{ extreme point of } P$$
$$\alpha^T r^j \geq 0 \ \textit{ for all } r^j \textit{ extreme} \quad \textit{ray of } P \ \}.$$

Proof. Let $Q' = \{\alpha \in \mathbb{R}^n : \ \alpha^T x^k \geq 1, \ \forall x^k$ extreme point of $P, \ \alpha^T r^j \geq 1, \ \forall r^j$ extreme ray of $P\}$. Suppose $\bar{\alpha} \in Q$. For every x^k extreme point of P and r^j extreme ray of P, we have $\bar{\alpha}^T(x^k + \mu r^j) \geq 1$ for all $\mu \geq 0$. This implies $\bar{\alpha}^T x^k \geq 1$ and $\bar{\alpha}^T r^j \geq 0$. Hence $\bar{\alpha} \in Q'$, so $Q \subseteq Q'$. Conversely if $\alpha \in Q'$ and $x \in P$, then $x = \sum_k \lambda_k x^k + \sum_j \mu_j r^j$ for some λ, μ satisfying $\sum_k \lambda_k = 1, \ \lambda_k \geq 0, \ \mu_j \geq 0$. Hence $\alpha^T x = \sum_k \lambda_k(\alpha^T x^k) + \sum_j \mu_j(\alpha^T r^j) \geq 1$. Therefore $Q' \subseteq Q$. □

Proposition 3 gives a set of constraints describing Q and we know from linear programming theory that all facet-defining inequalities for Q are part of these constraints (modulo scalar multiplication). The description may also include non-facet-defining, hence redundant, constraints. However, Proposition 4 shows that all constraints of the form $\alpha^T x^k \geq 1$, where x^k is a vertex of P, are facet-defining for Q.

Proposition 4. *The facet-defining inequalities of Q are*

(a). $\alpha^T x^k \geq 1$ for all x^k extreme point of P

(b). $\alpha^T r^j \geq 0$ for all r^j extreme ray of P such that $r^j \notin \mathrm{cone}\{x : x$ extreme point of $P\}$.

Proof. (a). Let y be a vertex of P and let $P^y = \mathrm{conv}\{x : x \text{ is a vertex of P }, x \neq y\} + \mathrm{recc}(P)$. Because $\mathrm{recc}(P^y) = \mathrm{recc}(P)$, P^y is also radial, and we denote its polar by Q^y. Obviously, $P^y \subsetneq P$, indeed $y \in P \setminus P^y$, and by Proposition 2, there exists an inequality $\bar{\alpha}^T x \geq 1$ that is valid for P^y and separates y. Thus $\bar{\alpha} \in Q^y$ while $\bar{\alpha} \notin Q$, proving that $Q^y \neq Q$. Therefore, all the inequalities of the form $\alpha^T y \geq 1$ for y extreme point of P are necessary to the description of Q, and are hence facet-defining for Q.

(b). Let $P^X = \mathrm{conv}\{x : x \text{ is a vertex of } P\} + \mathrm{cone}\{x : x \text{ is a vertex of } P\}$. Since P is radial, $P^X \subseteq P + \mathrm{cone}(P)$ and as noted earlier, $P = P + \mathrm{cone}(P)$, so $P^X \subseteq P$. Let t be an extreme ray of P such that $t \notin \mathrm{cone}\{x : x \text{ extreme point of } P\}$, i.e. $t \notin \mathrm{recc}(P^X)$, and let $P^t = P^X + \mathrm{cone}\{r : r \text{ is an extreme ray of } P, r \neq t\}$. Because t is an extreme ray of P, it can not be expressed as a conic combination of other rays of P, so $t \notin \mathrm{recc}(P^t)$. By construction, P^t is radial and we denote its polar by Q^t. Furthermore, $\mathrm{recc}(P^t) \subseteq \mathrm{recc}(P)$ hence $P^t \subseteq P$. Let w be an arbitrary vertex of P. As $t \notin \mathrm{recc}(P^t)$, there exist $M \in \mathbb{R}_+$ sufficiently large such that $z = w + Mt$ does not belong to P^t, while by construction it belongs to P. By Proposition 2, there exists an inequality $\tilde{\alpha}^T x \geq 1$ that is valid for P^t and separates z. Thus $\tilde{\alpha} \in Q^t$ while $\tilde{\alpha} \notin Q$, proving that $Q^t \neq Q$. Therefore, all inequalities of the form $\alpha^T t \geq 0$ for t extreme ray of P such that $t \notin \mathrm{cone}\{x : x \text{ vertex of } P\}$ are necessary to the description of Q, and are hence facet-defining for Q. □

One elegant property of radial polyhedra, which they share with full-dimensional polytopes [26], is a simple duality relationship between them and their polar. Proposition 5 and Proposition 6 establish this duality.

Proposition 5. *The polar Q of a radial polyhedron P is a radial polyhedron.*

Proof. (a). Since $0 \notin P$, by Proposition 2, there exist a valid inequality $\bar{\alpha}^T x \geq 1$ for P that separates 0, hence $\bar{\alpha} \in Q$, showing that Q is not empty. (b). Since P is not empty, $0^T x \geq 1$ is not a valid inequality for P, thus $0 \notin Q$. (c). By Proposition 3, Q is a polyhedron. Let $\alpha \in Q$, we know that $\alpha^T x \geq 1$ for all $x \in P$. Then for all $x \in P$, $(\mu\alpha)^T x = \alpha^T(\mu x) \geq 1$, since $\mu x \in P$. Thus $\mu\alpha \in Q$. □

Proposition 6 ([26] Proposition 5.4)**.** *The polar of Q is P.*

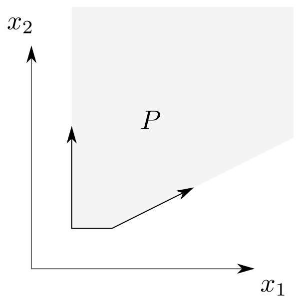

Figure 12: Example radial set P

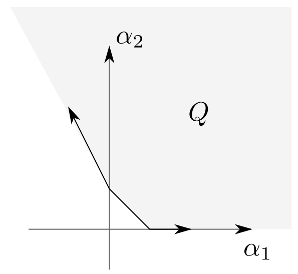

Figure 13: Example polar Q

Proof. By Proposition 5, Q is a radial polyhedron, thus its polar can be defined as in Definition 8. Let $\overline{P} = \{y \in \mathbb{R}^n : \ y^T\alpha \geq 1, \text{ for all } \alpha \in Q\}$ be the polar of Q. If $x \in P$, then $\alpha^T x \geq 1$, for all $\alpha \in Q$. Thus $x \in \overline{P}$, so $P \subseteq \overline{P}$. Now let $y \notin P$. By Proposition 2, there exists a valid inequality $\alpha^T x \geq 1$ of P such that $\alpha^T y < 1$. Since $\alpha \in Q$, $y \notin \overline{P}$, so $\overline{P} \setminus P = \emptyset$. $\square$

Corollary 3. *The facet-defining inequalities of P are*

(a). $\alpha^T x \geq 1$ for all α extreme point of Q

(b). $\beta^T x \geq 0$ for all β extreme ray of Q such that $\beta \notin \text{cone}\{\alpha : \alpha \text{ extreme point of } P\}$.

The following example illustrates the properties that we established in this section.

Example 1. *Let $P \subseteq \mathbb{R}^2$ be given by (Figure 12)*

$$P = \text{conv}\left(\left(\begin{array}{c} 2 \\ 1 \end{array}\right), \left(\begin{array}{c} 1 \\ 1 \end{array}\right)\right) + \text{cone}\left(\left(\begin{array}{c} 2 \\ 1 \end{array}\right), \left(\begin{array}{c} 0 \\ 1 \end{array}\right)\right).$$

It is easy to verify that P is radial. From the vertices and extreme rays of P immediately follows a description of its polar Q in terms the constraints (Figure 13)

$$\begin{aligned} Q = \{(\alpha_1, \alpha_2) \in \mathbb{R}^2 : 2\alpha_1 + \alpha_2 &\geq 1 && (27)\\ \alpha_1 + \alpha_2 &\geq 1 && (28)\\ 2\alpha_1 + \alpha_2 &\geq 0 && (29)\\ \alpha_2 &\geq 0\}. && (30) \end{aligned}$$

By optimizing over Q, we can obtain vertices and extreme rays of Q. In our small example, we can observe that

$$Q = \text{conv}\left(\left(\begin{array}{c} 1 \\ 0 \end{array}\right), \left(\begin{array}{c} 0 \\ 1 \end{array}\right)\right) + \text{cone}\left(\left(\begin{array}{c} 1 \\ 0 \end{array}\right), \left(\begin{array}{c} -1 \\ 2 \end{array}\right)\right)$$

and that (27), (28) and (30) are facet-defining for Q while (29) is not (it is strictly dominated by (27)). Indeed, the corresponding extreme ray of P

$$\left(\begin{array}{c} 2 \\ 1 \end{array}\right) \in \text{cone}\left(\left(\begin{array}{c} 2 \\ 1 \end{array}\right), \left(\begin{array}{c} 1 \\ 1 \end{array}\right)\right).$$

Conversely, the vertices and extreme rays of Q yield a constraint description of P

$$\begin{aligned} P = \{(x_1, x_2) \in \mathbb{R}^2 : \quad x_1 \qquad &\geq 1 && (31)\\ x_2 &\geq 1 && (32)\\ x_1 \qquad &\geq 0 && (33)\\ -x_1 + 2x_2 &\geq 0\} && (34) \end{aligned}$$

where (31)*,* (32) *and* (34) *are facet-defining for* Q *while* (33) *is not (it is strictly dominated by* (31)*). Again, the corresponding ray of* Q

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \in \operatorname{cone}\left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right).$$

References

[1] Tobias Achterberg, Thorsten Koch, and Alexander Martin. MIPLIB 2003. *Operations Research Letters*, 34(4):361–372, 2006.

[2] Kent Andersen, Quentin Louveaux, and Robert Weismantel. Mixed-integer sets from two rows of two adjacent simplex bases. *Mathematical Programming*, 124:455–480, 2010.

[3] Kent Andersen, Quentin Louveaux, Robert Weismantel, and Laurence Wolsey. Cutting planes from two rows of a simplex tableau (extended version), 2006. Working Paper available on `http://orbi.ulg.ac.be/handle/2268/82794`.

[4] Kent Andersen, Quentin Louveaux, Robert Weismantel, and Laurence Wolsey. Inequalities from two rows of a simplex tableau. In Matteo Fischetti and David Williamson, editors, *Integer Programming and Combinatorial Optimization*, volume 4513 of *Lecture Notes in Computer Science*, pages 1–15. Springer Berlin / Heidelberg, 2007.

[5] Kent Andersen, Christian Wagner, and Robert Weismantel. On an analysis of the strength of mixed-integer cutting planes from multiple simplex tableau rows. *SIAM Journal on Optimization*, 20(2):967–982, 2009.

[6] Alexander Barvinok. A polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed. *Math. Oper. Res.*, 19(4):769–779, 1994.

[7] Alexander Barvinok. *Integer Points in Polyhedra*. Zurich Lectures in Advanced Mathematics. European Mathematical Society, 2008.

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

[9] Amitabh Basu, Pierre Bonami, Gérard Cornuéjols, and François Margot. Experiments with two-row cuts from degenerate tableaux. *INFORMS Journal on Computing*, 2011.

[10] Amitabh Basu, Pierre Bonami, Gérard Cornuéjols, and François Margot. On the relative strength of split, triangle and quadrilateral cuts. In *SODA*, pages 1220–1229. SIAM, 2009.

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

[12] Valentin Borozan and Gérard Cornuéjols. Minimal valid inequalities for integer constraints. *Math. Oper. Res.*, 34(3):538–546, 2009.

[13] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. A geometric perspective on lifting. *Oper. Res.*, 59:569–577, May 2011.

[14] Gérard Cornuéjols and François Margot. On the facets of mixed integer programs with two integer variables and two constraints. *Math. Program.*, 120(2):429–456, 2009.

[15] Santanu S. Dey, Andrea Lodi, Andrea Tramontani, and Laurence A. Wolsey. Experiments with two row tableau cuts. In Friedrich Eisenbrand and F. Bruce Shepherd, editors, *Integer Programming and Combinatorial Optimization, 14th International Conference, IPCO 2010, Lausanne, Switzerland, June 9-11, 2010. Proceedings*, volume 6080 of *Lecture Notes in Computer Science*, pages 424–437. Springer, 2010.

[16] Santanu S. Dey and Quentin Louveaux. Split rank of triangle and quadrilateral inequalities. *Mathematics of Operations Research*, 36(3):432–461, 2011.

[17] Santanu S. Dey and Laurence A. Wolsey. Lifting integer variables in minimal inequalities corresponding to lattice-free triangles. In Andrea Lodi, Alessandro Panconesi, and Giovanni Rinaldi, editors, *Integer Programming and Combinatorial Optimization, 13th International Conference, IPCO 2008, Bertinoro, Italy, May 26-28, 2008, Proceedings*, volume 5035 of *Lecture Notes in Computer Science*, pages 463–475. Springer, 2008.

[18] Santanu S. Dey and Laurence A. Wolsey. Constrained infinite group relaxations of MIPs. CORE Discussion Papers 2009033, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), May 2009.

[19] Daniel G. Espinoza. Computing with multi-row Gomory cuts. *Operations Research Letters*, 38(2):115 – 120, 2010.

[20] Ricardo Fukasawa and Oktay Günlük. Strengthening lattice-free cuts using non-negativity. *Discrete Optimization*, 8(2):229 – 245, 2011.

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

[22] R. E. Gomory and E. L. Johnson. Some continuous functions related to corner polyhedra, part II. *Mathematical Programming*, 3:359–389, 1972.

[23] Ralph E. Gomory. Some polyhedra related to combinatorial problems. *Linear Algebra and its Applications*, 2(4):451 – 558, 1969.

[24] Warwick Harvey. Computing two-dimensional integer hulls. *SIAM J. Comput*, 28(6):2285–2299, 1999.

[25] L. Lovàsz. Geometry of numbers and integer programming. *Proc. Math. Appl. Jap. Ser. 6*, pages 177–201, 1989.

[26] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization.* John Wiley & Sons, 1988.