UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA MATEMÁTICA

SOME EXPERIMENTS ON SEPARATION WITH MULTI-ROW CUTS

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN GESTIÓN DE OPERACIONES

MEMORIA PARA OPTAR AL TITULO DE INGENIERO CIVIL EN MATEMÁTICAS

FELIPE SERRANO MUSALEM

PROFESOR GUÍA:
DANIEL GONZALEZ ESPINOZA

MIEMBROS DE LA COMISIÓN:
JOSE RAFAEL CORREA HAEUSSLER
MARTIN IGNACIO MATAMALA VASQUEZ
MARCOS GOYCOOLEA GUZMAN

SANTIAGO DE CHILE
MARZO 2012

Resumen

Desde que Andersen et al. introdujeron los "*multi-row cuts*", ha habido mucho interés en ellos. Sin embargo, la cantidad de estudios computacionales ha sido limitada. La mayor parte de las investigaciones se centran en los *multi-row cuts* que pueden deducirse usando dos filas, relajando las cotas de las variables o, al menos, de un tipo de las variables. En este último caso, se relaja la integralidad de las variables no básicas. Asimismo, la mayor parte de las investigaciones busca sólo la separación exacta.

En esta tesis, utilizamos un enfoque numérico que nos permite explorar relajaciones más complejas e introducimos un esquema de separación aproximada, que pueda ser útil desde el punto de vista de la implementación práctica. También, incorporamos criterios sencillos para aprovechar la integralidad de las variables no básicas. Es importante asegurar que los cortes generados sean válidos, por lo que incluimos métodos que buscan la estabilidad numérica, minimizando la creación de cortes falsos.

Una vez obtenido el tableau, buscamos "*deep cuts*", que definimos como cortes ($\alpha x \geq 1$) que minimizan $||\alpha||$. Para encontrarlos, resolvemos el dual del problema de minimización mediante generación de columnas.

Probamos las normas ℓ^1 y ℓ^2, y la no-linealidad es resuelta aplicando la aproximación del cono de segundo orden de Ben-Tal and Nemirovsky. Para acelerar el proceso, buscamos representaciones violadas de algunos puntos fijos.

Igualmente, probamos diferentes criterios para la selección de filas: al azar, mayor producto punto y menor producto punto. Para obtener una idea más acabada sobre la capacidad de los *multi-rows cuts*, utilizamos todas las combinaciones posibles de filas para generar cortes, pero sin agregación de filas. Comparamos esto último con los resultados de Balas respecto al *split closure*.

Finalmente, analizamos el impacto en el nodo raíz de estos cortes sobre la librería MIPLIB3 y elegimos una buena configuración de las distintas opciones para probar en *branch and bound*.

Abstract

Since Andersen et al. there has been a lot of interest in multi-row cuts. However, computational study has been limited. Most research considers multi-row cuts deduced from only 2 rows and they use bounds on none or only on one type of variables, always relaxing integrality of non-basic variables when bounds are taken into account. Also, most applications aim to exact separation as well as using fixed convex lattice free bodies to separate.

In this work we try a numerical approach that allows us to look into more complex relaxations and we introduce an approximated separation hoping for a practical implementation to be possible. Extensively numerical analysis has been done to ensure numerical stability and minimize the creation of false cuts. Also, we incorporate some simple forms of taking advantage of integrality of non basic variables. Once the rows of a tableau are obtained we search for a "deep cut" which we understand as a cut ($\alpha x \geq 1$) that minimizes $||\alpha||$. To find it, we solve the dual using a column generation approach.

We tested both, ℓ^1-norm and ℓ^2-norm, where the latter one is treated using Ben-Tal and Nemirovsky approximation of the second order cone. In order to speed up the process we seek for violated representations of fixed points.

Different criteria for row selection are tested: random selection, largest dot product and smallest dot product. To give a more complete idea about the strength of multi-rows cuts, we also generated all possible cuts using all combination of rows, but without aggregation of rows. We compare this to Balas computations of the split closure.

As for the experiments done in this work, we analyze the impact in the root node of the procedure (using various rounds) over MIPLIB3. Also, we select a good configuration and test its performance in branch and bound.

Contents

1 Introduction

The most powerful approach to solve general MIP we have today is branch and cut. Although branch and bound has been around since its introduction by Land and Doig in 1960, it was not until the implementation of cuts that we were able to solve big and difficult problems. Maybe the most important family of general purpose cuts are the GMI [8]. These cuts are obtained after studying mixed integer sets with only one constraint.

For the pure integer case, it is known that a cutting plane algorithm converges when GMI cuts are used, but this is not true for the mixed integer problems. Furthermore, in [10] a three variable mixed integer problem, with infinite Chvatal-Gomory rank, is presented. This means that simply adding cuts derived from one row relaxation of the tableau (in particular, GMI cuts) will not solve the problem. Recently, Anderson et al. [3] showed that a single cut derived from a two row relaxation is enough to solve this problem. However, the example was generalized by Li and Richards [19], so this behavior repeats at higher dimension.

Anderson et al. [3] considered following mixed integer set

$$X = \left\{ (x,s) \in \mathbb{Z}^q \times \mathbb{R}^n_+ : x = f + \sum_{j \in N} r^j s_j \right\}$$

where all data is rational and $f \notin \mathbb{Z}^q$.

They studied the case $q = 2$ and geometrically characterized all facets of $\text{conv}(X)$, showing a relationship between facets and maximal lattice free convex sets in $\mathbb{R}^2$.

Gomory and Johnson [17] suggested the semi-infinite relaxation of X:

$$\begin{aligned} & x = f + \sum_{r \in \mathbb{Q}^q} r s_r \\ & x \in \mathbb{Z}^q \\ & s \geq 0 \\ & s \text{ has finite support} \end{aligned}$$

Borozan and Cornuéjols [9] show that for the infinite relaxation of X there is a one to one correspondence between minimal valid inequalities and maximal lattice free convex sets in $\mathbb{R}^q$ that contain f in their interior and that minimal valid functions for the semi-infinite relaxation are non-negative, piecewise linear, positively homogeneous and convex. Using these results, Cornuéjols and Margot [11] determine conditions that valid inequalities of X must satisfy to be facets when

$q = 2$. Espinoza [14] used the results of [9] in a computational study of multi-row cuts with as much as $q \leq 15$ rows, where he fixes the lattice free convex set used to separate.
More recently, different relaxations have been taken into account. In [2] Andersen et al. extended their previous work by considering upper bounds on the continuous variables of X where $q = 2$ and Fukasawa and Günlük [15] considered nonnegativity on integral variables. Dey and Wolsey [13] used the integrality of some nonbasic variables to lift minimal inequalities derived from lattice free triangles. Dey et al. [12] performed a computational study of cuts based on lattice free triangles ($q = 2$), using bounds and lifting integer nonbasic variables. Louveaux and Poirrier [20] proposed an algorithm that does not assume the shape of the lattice free set used to separate, instead, they search deep cuts for $\text{conv}(X)$ with $q = 2$.

In this thesis, we investigate a column generation framework that enables us to deduce multi-row cuts from any number of rows and to explore more complex relaxations than the ones already in the literature. We will seek cuts that minimize their ℓ^1-norm or ℓ^2-norm. In addition, several options can be set:

- bounds on none, basic, nonbasic or both variables
- consider nonbasic integer variables
- different criteria for row selection

This way we can computationally explore what has only been explored theoretically. In particular, we can test the importance on bounds when using multi row cuts, something that has not been tested, even though it gives a stronger relaxation. This also let us work with more dimensions, where most of the results valid for two row relaxations have not been generalized.
Aside from testing the importance of different relaxations, we are trying to obtain an effective way to derive multi-row cuts. An important aspect is the time used in the cut generation process, so it is necessary to have a way of dealing with high dimensional relaxations.
With this in mind, we propose an approximated separation that, instead of fixing the shape of the lattice free set, fixes the rays r^j. We try to determine whether the number of rays used in the approximation is important, how much we can gain in time and how much we lose in quality.
We perform experiments on MIPLIBs 3 [7], 2003 [1] and 2010 [18], comparing the performance on the root node between different types of relaxations and

against CPLEX alone. We try to answer the following questions. Does considering integrality of nonbasic variables give us better cuts? How do three simple row selecting criteria influence the process? Is the approximation scheme a viable way to deduce cuts, either by time issues and/or quality? Which are the best relaxations to derive cuts?

The thesis is organized as follows. Section 2 introduces some definitions and notations, as well as the general framework. In section 3 we show how to deal with high dimensional relaxation and the concept of *transformation* is introduced, as well as the approximation. Section 4 shows how to build the relaxation. Section 5 discuses some numerical issues and a way to improve the overall performance. Finally, section 6 presents computational results.

2 Background & Framework

We consider a general mixed integer problem (MIP),

$$\begin{aligned} \min \quad & c^t x \\ \text{s.t} \quad & \\ & Ax = b \\ & x \geq 0 \\ & x_i \in \mathbb{Z} \text{ for } i \in I \end{aligned}$$

where all data is rational and I is the set of integer variables. Denote the linear relaxation of the MIP by LP and let x^* be a basic feasible solution with basic variables B and nonbasic variables N. We also denote B and N the sub-matrices of A associated with basic and nonbasic variables respectively.
We can write the problem as follows

$$x_i + \sum_{j \in N} \bar{a}_{ij} x_j = \bar{b}_i \quad \forall i \in B$$

where $\bar{b}_i = \left(B^{-1}b\right)_i$ and $\bar{a}_{ij} = \left(B^{-1}N\right)_{ij}$. We are interested in multi-row cuts, so we select $q > 1$ rows $i_1, \ldots, i_q \in B \cap I$ such that $\left(\bar{b}_{i_1}, \ldots, \bar{b}_{i_q}\right) \notin \mathbb{Z}^q$.
Let us call,

$$\begin{aligned} f &= \left(\bar{b}_{i_1}, \ldots, \bar{b}_{i_q}\right)^t \\ r^j &= -\left(\bar{a}_{i_1 j}, \ldots, \bar{a}_{i_q j}\right)^t \end{aligned}$$

Finally, the mixed integer set that we will study is

$$X = \left\{ (x,s) \in \mathscr{S} \times \mathbb{R}^n_+ \colon x = f + \sum_{j \in N} r^j s_j,\ s_j \leq u_j \right\}$$

where $\mathscr{S} = \{x \in \mathbb{Z}^q \colon a \leq x \leq b\}$, $u_j \in \mathbb{R} \cup \{+\infty\}$, $a \in (\mathbb{R} \cup \{-\infty\})^q$ and $b \in (\mathbb{R} \cup \{+\infty\})^q$. We also write N for $\{1, \ldots, n\}$.
Let

$$R_f\left(r^1, \ldots, r^n; u; \mathscr{S}\right) = \left\{ s \in \mathbb{R}^n_+ \colon f + \sum_{j \in N} r^j s_j \in \mathscr{S},\ s_j \leq u_j \right\} \tag{1}$$

When context allows it, we will write R_f instead of $R_f\left(r^1, \ldots, r^n; u; \mathscr{S}\right)$. Note that R_f is the projection of X to $\mathbb{R}^n$. Our objective is to find valid inequalities for $\text{conv}(X)$ that cut off the point $(f, 0)$. Let us start with a proposition that gives us the form of these cuts.

Proposition 2.1. *Every non-trivial valid inequality for X that cuts off the point* $(f,0)$ *can be written as*

$$\sum_{i\in N} \alpha_i s_i \geq 1$$

Proof. Let (π,π_0) be a valid inequality for X, i.e., $\sum_{i=1}^{q} \pi_i x_i + \sum_{j\in N} \pi_j s_j \geq \pi_0$ is satisfied for every $(x,s) \in X$. As $x_i = f_i + \sum_{j\in N} r_i^j s_j$, the last inequality is equivalent to $\sum_{j\in N} \theta_j s_j \geq \theta_0$ where $\theta_0 = \pi_0 - \sum_{i=1}^{q} \pi_i f_i$ and $\theta_j = \pi_j + \sum_{i=1}^{q} \pi_i r_i^j$. For $(f,0)$ to satisfy this inequality we would need that $0 \geq \theta_0$, then $\theta_0 > 0$. If we let $\alpha_j = \frac{\theta_j}{\theta_0}$ we obtain the desired form. □

We are going to ask our cuts to displace the point $s = 0$ the most. The distance between 0 and the hyperplane $\sum_{j\in N} \alpha_j s_j = 1$ is $\frac{1}{||\alpha||_2}$. In order to maximize this distance, we need to find cuts that have a small norm.

Definition 2.2. *A cut* α *is a* ***deep cut*** *if it solves the problem*

$$\begin{aligned} &\min && ||\Gamma\alpha|| \\ &s.t \\ & && \sum_{j\in N} \alpha_j s_j \geq 1,\ \forall s \in R_f \end{aligned} \tag{2}$$

where Γ *is a* $n \times n$ *diagonal matrix with non-negative entries and the norm can be the* ℓ^1*-norm or* ℓ^2*-norm.*

The Γ matrix is a weight matrix that will only appear when some transformation is applied (see the next section). If no transformation is applied, this matrix is the identity.
Andersen et al. showed in [3] that when no bounds are present, the cut's coefficients α_j must be non-negative for every j, but when we consider bounds on the x variables, Fukasawa and Günlük showed an example where some coefficients can be negative (see [15]). The same is true when bounds on s are considered (see [2]). For deep cuts, we have the following proposition.

Proposition 2.3. *Let* α *be the optimal solution of (2), then* $\alpha_j \geq 0$ *for all* $j \in N$

Proof. Suppose that α has a negative component, say $\alpha_j < 0$. Then α^* given by $\alpha_i^* = \alpha_i,\ i \neq j$ and $\alpha_j^* = 0$ is a feasible point that has lower objective value since all s are positive and Γ has non-negative entries. This contradicts the minimality of α. □

This implies that the bounds $\alpha \geq 0$ can be added to (2). Note that this is true for any norm.
Before we go into the column generation approach, we will introduce some sets that where defined in [3].

Definition 2.4. *Let X_{LP} be the linear relaxation of X, i.e.,*

$$X_{LP} = \left\{ (x,s) \in \bar{\mathscr{S}} \times \mathbb{R}^n_+ \colon x = f + \sum_{j \in N} r^j s_j,\ s_j \leq u_j \right\}.$$

where $\bar{\mathscr{S}} = \{x \in \mathbb{R}^q \colon a \leq x \leq b\}$.
Given $\alpha \in \mathbb{R}^n_+$, let

$$L_\alpha = \left\{ x \in \bar{\mathscr{S}} \colon \text{there exist } s \in \mathbb{R}^n \text{ such that } (x,s) \in X_{LP} \text{ and } \sum_{j \in N} \alpha_j s_j \leq 1 \right\}.$$

In other words, L_α is the projection to the R^n of the points in X_{LP} that violate the inequality $\sum_{j \in N} \alpha_j s_j \geq 1$.

The following results apply when no bounds are present, so we will work with $R_f = R_f\left(r^1, \ldots, r^n; \infty; \mathbb{Z}^q\right)$.
Let $\alpha \in \mathbb{R}^n_+$, $N_+ = \left\{ j \in N \colon \alpha_j > 0 \right\}$ and $v^j = f + \frac{r^i}{\alpha_j}$ for $j \in N_+$. From [3] we know that

$$L_\alpha = \operatorname{conv}\left(f \cup \left\{ v^j \right\}_{j \in N_+} \right) + \operatorname{cone}\left(\left\{ r^j \right\}_{j \in N \setminus N_+} \right).$$

Notice that α is a valid inequality for R_f if and only if the interior of L_α is lattice free.

Observation 2.5. *For a general R_f we can still define the set L_α. It is straightforward to see that α is a valid inequality for R_f if and only if the interior of L_α is $\mathscr{S}$ free, i.e. it does not contain any point of $\mathscr{S}$.*

Proposition 2.6. *Let α be the optimal solution of (2). Then for every $j \in N_+$, v^j belongs to the boundary of L_α.*

Proof. Choose any $j \in N_+$. First, lets show that the half-line $\left\{ f + \lambda r^j \colon \lambda \in \mathbb{R}_+ \right\}$ intersects the boundary of L_α. To prove this, assume that the half-line does not intersect the boundary. Take any $0 < \varepsilon < \alpha_j 0$ and define $\tilde{\alpha}$ as $\tilde{\alpha}_i = \alpha_i$ if $i \neq j$, $\tilde{\alpha}_j = \alpha_j - \varepsilon$ and $\tilde{v}^j = f + \frac{r^j}{\tilde{\alpha}_j}$.

We have that $\tilde{\alpha}$ is a cut since $L_{\tilde{\alpha}} = L_\alpha$ and L_α does not contain any integer point in its interior. Also $||\tilde{\alpha}|| < ||\alpha||$, which contradicts the optimality of α.
Let us prove now that v^j belongs to the boundary of L_α. If not, there exist $\varepsilon > 0$ such that $\tilde{v}^j = f + \frac{r^j}{\alpha_j - \varepsilon} \in L_\alpha$. Defining $\tilde{\alpha}$ as above and using the same argument we conclude. □

2.1 Column generation

In general, the number of constraints of (2) is very large (possibly infinite). As it is easier for solvers to deal with many variables instead of many constraints, we apply a column generation approach to solve the dual of (2).

2.1.1 The ℓ^1-norm case

We can restate our primal problem (2). Let $\gamma_j = \Gamma_{jj}$, as α_j and γ_j are non-negative for every $j \in N$, the primal can be written as

$$\begin{aligned} \min \quad & \sum_{j \in N} \gamma_j \alpha_j \\ \text{s.t.} \quad & \\ & \sum_{j \in N} \alpha_j s_j \geq 1 \ \forall s \in R_f \\ & \alpha_j \geq 0 \ \forall j \in N \end{aligned} \tag{3}$$

Define $\gamma = \left(\gamma_j\right)_{j \in N}$. Then the dual in vector notation is

$$\begin{aligned} \max \quad & \sum_{s \in R_f} \beta_s \\ \text{s.t} \quad & \\ & \sum_{s \in R_f} s\beta_s \leq \gamma \\ & \beta_s \geq 0 \end{aligned} \tag{4}$$

The objective of the column generation is to solve (4). For this, we start with $R'_f = \emptyset$ instead of R_f. The dual values of (4) give us a cut candidate π, initially 0, that is tested in the **subproblem**:

$$
\begin{aligned}
\min \quad & \sum_{j \in N} \pi_j s_j \\
\text{s.t} \quad & \\
& f + \sum r^i s_i = x \\
& 0 \leq s \leq u \\
& a \leq x \leq b \\
& x \in \mathbb{Z}^q
\end{aligned}
\tag{5}
$$

If the solution of (5) is less than 1, then there is a point that violates the cut. We add that point to R'_f and repeat. If the solution is greater than 1, then it is a valid cut.
As usual, we will call (4), with R'_f instead of R_f, the **master** problem.

2.1.2 The ℓ^2-norm case

In this case, the primal problem is

$$
\begin{aligned}
\min \quad & ||\Gamma \alpha||_2 \\
\text{s.t.} \quad & \\
& \textstyle\sum_{j \in N} \alpha_j s_j \geq 1 \ \forall s \in R_f \\
& \alpha_j \geq 0 \ \forall j \in N
\end{aligned}
\tag{6}
$$

Since Γ is a diagonal matrix, the dual problem is

$$
\begin{aligned}
\max \quad & \sum_{s \in R_f} \beta_s \\
\text{s.t} \quad & \\
& \sum_{s \in R_f} s\beta_s \leq \Gamma y \\
& ||y||_2 \leq 1 \\
& y \geq 0 \\
& \beta_s \geq 0
\end{aligned}
\tag{7}
$$

To linearize the constraint $||y||_2 \leq 1$ we use Ben-Tal and Nemirovski approximation [6]. Basically, the method consists on repeatedly using the approximation of the cone $L^2 = \left\{(y,t) \in \mathbb{R}^2 \times \mathbb{R} \colon \ ||y||_2 \leq t\right\}$ given by

$$
\begin{aligned}
&\xi^0 \geq |y_1| \\
&\eta^0 \geq |y_2| \\
&\xi^j = \cos\left(\frac{\pi}{2^{j+1}}\right)\xi^{j-1} + \sin\left(\frac{\pi}{2^{j+1}}\right)\eta^{j-1} \\
&\eta^j \geq \left|-\sin\left(\frac{\pi}{2^{j+1}}\right)\xi^{j-1} + \cos\left(\frac{\pi}{2^{j+1}}\right)\eta^{j-1}\right|, \quad j = 1,\ldots,\nu \\
&\xi^\nu \leq t \\
&\eta^\nu \leq \tan\left(\frac{\pi}{2^{\nu+1}}\right)\xi^\nu
\end{aligned}
$$

where ν is a parameter that controls the quality of the approximation. The column generation scheme is the same as above, except that the cut candidate π is obtained from the dual values associated with the constraints involving β_s variables only.

3 Dealing with the high dimension of R_f

The column generation method presented so far, will require at least n iterations, hence the process can be very slow if n is too big. In this section we present a general transformation idea that allows us to work with less rays. Then, we show how to remove parallel rays, proving that in some cases we obtain the same solution and showing that in others we will just get a non optimal solution. Finally, we present the approximation scheme.

3.1 Transformations

Finding a valid inequality for R_f can be very expensive. Here we will present a simple way to transform valid inequalities from a simpler set $\hat{R}_f$ to the original more complex set R_f.

Definition 3.1. *Let $R_f = R_f\left(r^1,\dots,r^n;u;\mathscr{S}\right)$ and $\hat{R}_f = R_f\left(q^1,\dots,q^m;\hat{u};\hat{\mathscr{S}}\right)$. A linear function $T: R_f \to \hat{R}_f$ will be called a **transformation**.*

As it is usual, we will refer to the matrix underlying T as T.

Proposition 3.2. *Let R_f and $\hat{R}_f$ be two sets as before, and $T: R_f \to \hat{R}_f$ a transformation. If $\hat{\alpha}$ is a cut for $\hat{R}_f$, then $T^t\hat{\alpha}$ is a cut for R_f.*

Proof. Let $s \in R_f$. As $Ts \in \hat{R}_f$ and $\hat{\alpha}$ is a cut for $\hat{R}_f$, then

$$\begin{aligned} 1 &\leq \hat{\alpha}\cdot Ts \\ &= \hat{\alpha}^t Ts \\ &= \left(T^t\hat{\alpha}\right)^t s \\ &= T^t\hat{\alpha}\cdot s \end{aligned}$$

As s is general, we conclude that $T^t\hat{\alpha}$ is a cut for R_f □

Now, instead of looking for deep cuts in R_f we can restrict to a subfamily of cuts, namely, cuts of the form $T^t\hat{\alpha}$. So, instead of solving (2), we solve

$$\begin{aligned} \min \quad & \left|\left|T^t\hat{\alpha}\right|\right| \\ \text{s.t} \quad & \\ & T^t\hat{\alpha}\cdot s \geq 1,\ \forall s \in R_f \\ & \hat{\alpha}_i \geq 0, \ \text{ for } i = 1,\dots,m \end{aligned}$$

If T is onto, this problem is equivalent to the following.

$$\begin{aligned} \min \quad & \left|\left|T^t\hat{\alpha}\right|\right| \\ \text{s.t} \quad & \\ & \hat{\alpha}\cdot\hat{s} \geq 1,\ \forall \hat{s}\in\hat{R}_f \\ & \hat{\alpha}_i \geq 0, \ \text{ for } i=1,\ldots,m \end{aligned} \tag{8}$$

Even though T is not necessarily onto, we will still solve (8). Due to the type of transformations that will arise later, we will always be able to find a diagonal matrix Γ such that $||T^t\hat{\alpha}|| = ||\Gamma\hat{\alpha}||$.

3.2 Removing parallel rays

When no upper bounds on s variables are present, it is easy to see that one can forget about rays pointing in the same direction.

Proposition 3.3. *Let $j_1, j_2 \in N$, $j_1 \neq j_2$ be such that $r^{j_1} = \lambda r^{j_2}$ for some $\lambda \geq 0$. If α solves (2) without upper bounds on s variables, then $\alpha_{j_1} = \lambda\alpha_{j_2}$*

Proof. Suppose that $\alpha_{j_1} < \lambda\alpha_{j_2}$. We just have to prove that α^* given by

$$\begin{aligned} \alpha^*_i &= \alpha_i, \quad i \neq j_2 \\ \alpha^*_{j_2} &= \frac{\alpha_{j_1}}{\lambda} \end{aligned}$$

is a feasible point of (2). Let $s \in R_f$ and $N' = N\setminus\{j_1, j_2\}$, then

$$\begin{aligned} f + \sum_{j\in N'} s_j r^j + s_{j_1} r^{j_1} + s_{j_2} r^{j_2} &= f + \sum_{j\in N'} s_j r^j + s_{j_1} r^{j_1} + \frac{s_{j_2}}{\lambda}\lambda r^{j_2} \\ &= f + \sum_{j\in N'} s_j r^j + \left(s_{j_1} + \frac{s_{j_2}}{\lambda}\right) r^{j_1} \end{aligned}$$

Hence, s^* given by $s^*_j = s_j$ if $j \in N'$, $s^*_{j_1} = s_{j_1} + \frac{s_{j_2}}{\lambda}$ and $s^*_{j_2} = 0$, belongs to R_f.

Given that α is a valid inequality,

$$\begin{aligned}
1 &\leq \sum_{j\in N} s_j^* \alpha_j \\
&= \sum_{j\in N'} s_j \alpha_j + \left(s_{j_1} + \frac{s_{j_2}}{\lambda}\right)\alpha_{j_1} \\
&= \sum_{j\in N'} s_j \alpha_j^* + s_{j_1}\alpha_{j_1} + \frac{s_{j_2}}{\lambda}\alpha_{j_1} \\
&= \sum_{j\in N'} s_j \alpha_j^* + s_{j_1}\alpha_{j_1}^* + s_{j_2}\alpha_{j_2}^* \\
&= \sum_{j\in N} s_j \alpha_j^*
\end{aligned}$$

we conclude that α^* is feasible for (2). For the other case, just notice that if we change j_1 with j_2 and divide by λ we recover the first case. □

Our objective is to remove rays pointing in the same direction and show that, in some cases, the cuts we generate will not be affected by this. Let us start with a few definitions.

Definition 3.4. *We say that two rays, r^1 and r^2, are **parallel** (written $r^1 \parallel r^2$) if there exists a positive scalar λ, such that*

$$r^1 = \lambda r^2$$

We write $r^1 \nparallel r^2$ when the rays are not parallel.

Definition 3.5. *For every $j \in N$ we define N_j as the set of indices of rays parallel with j. That is*

$$N_j = \left\{i \in N\colon r^i \parallel r^j, i \neq j\right\}$$

Let $M \subseteq N$ be such that if $i, j \in M$, $i \neq j$, then $r^i \nparallel r^j$ and

$$M \cup \bigcup_{j\in M} N_j = N.$$

In other words, M is a maximal subset of indices, with respect to inclusion, of non-parallel rays.
For each $i \in N_j$, let $\lambda_i > 0$ be such that $r^j = \frac{r^i}{\lambda_i}$ and define T as $T(s) = \hat{s} \in \mathbb{R}^m$

where $\hat{s}_j = s_j + \sum_{i \in N_j} \lambda_i s_i$ and $m = |M|$. Let $R_f = R_f\left(\left(r^j\right)_{j\in N}; u; \mathscr{S}\right)$ and $\hat{R}_f = R_f\left(\left(r^j\right)_{j\in M}; T(u); \mathscr{S}\right)$. Clearly, $T\colon R_f \to \hat{R}_f$ is a transformation.

Whenever we remove parallel rays, we will be solving

$$\begin{aligned} &\min && \left|\left|T^t\alpha\right|\right| \\ &\text{s.t} && \\ &&& \sum_{j\in M} \alpha_j s_j \geq 1,\ \forall s \in \hat{R}_f \\ &&& \alpha_j \geq 0,\ \forall j \in M \end{aligned} \tag{9}$$

When solving with ℓ^1-norm, the diagonal matrix Γ given by $\Gamma_{jj} = 1 + \sum_{i\in N_j} \lambda_i$ for $j \in M$ satisfies

$$||\Gamma\alpha||_1 = \left|\left|T^t\alpha\right|\right|_1$$

For the ℓ^2-norm, we need $\Gamma_{jj} = \sqrt{1 + \sum_{i\in N_j} \lambda_i}$ instead.
In some cases the cut obtained in the transformed problem is the same as the cut obtained in the original problem.

Proposition 3.6. *Assume that s variables do not have upper bounds and that Γ is the identity matrix in (2). Let α^* be the solution of (9). Then $T^t\alpha^*$ solves (2).*

Proof. First notice that by Proposition 3.2, $T^t\alpha^*$ is a feasible point of (2). Let α be the optimal solution of (2).
Define $\hat{\alpha} \in \mathbb{R}^m$ by $\hat{\alpha}_j = \alpha_j$ for $j \in M$, i.e., $\hat{\alpha}$ is the projection to $\mathbb{R}^m$ of α.
We have that $\hat{\alpha}$ is a feasible point of (9). To see this, choose any $\hat{s} \in \hat{R}_f$ and let s be its extension by 0 to $\mathbb{R}^N$, then

$$f + \sum_{j\in M} \hat{s}_j r^j = f + \sum_{j\in N} s_j r^j.$$

This implies that $s \in R_f$, which implies that

$$\hat{\alpha}\cdot\hat{s} = \alpha\cdot s \geq 1.$$

From Proposition 3.3 it is easy to see that $T^t\hat{\alpha} = \alpha$. Finally, the optimality of α^* tells us that

$$\left|\left|T^t\alpha^*\right|\right| \geq \left|\left|T^t\hat{\alpha}\right|\right| = ||\alpha||$$

from which the proposition follows. □

When upper bounds on s are present this result is not generally true as shown in the following example.

Example 3.7. *Consider the rays* $r^1 = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$, $r^2 = \begin{pmatrix} -1 \\ -2 \end{pmatrix}$, $r^3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ *and* $r^4 = \begin{pmatrix} \frac{2}{3} \\ 0 \end{pmatrix}$. *Let* $N = 1,2,3,4$ *and* $R_f = R_f\left(r^1, r^2, r^3, r^4; \left(\infty, \infty, \infty, \frac{1}{12}\right); \mathbb{Z}^2\right)$ *where* $f = \frac{1}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$. *Using* ℓ^1*-norm, the problem that we want to solve is*

$$\begin{aligned} \min \quad & \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 \\ s.t \quad & \\ & \sum_{j \in N} \alpha_j s_j \geq 1, \ \forall s \in R_f \\ & \alpha_j \geq 0 \ \forall j \in N \end{aligned}$$

The point $\alpha_1 = \alpha_2 = 2$, $\alpha_3 = 0.72$ *and* $\alpha_4 = 0$ *is feasible, since* L_α *is lattice free as shown in Figure 1. Hence, the value of the optimal solution is at most* 4.72.
Now lets remove parallel rays. For this, let us choose $M = \{1,2,3\}$.
We have that $N_1 = N_2 = \emptyset$, $N_3 = 4$ *and* $\lambda_4 = \frac{2}{3}$, *then* $T(s) = \hat{s}$ *where*

$$\hat{s} = \left(s_1, s_2, s_3 + \frac{2}{3}s_4\right).$$

We can now write our new set, $\hat{R}_f = \left(r^1, r^2, r^3; (\infty, \infty, \infty); \mathbb{Z}^2\right)$ *and the problem we want to solve is*

$$\begin{aligned} \min \quad & \hat{\alpha}_1 + \hat{\alpha}_2 + \frac{5}{3}\hat{\alpha}_3 \\ s.t \quad & \\ & \sum_{j \in M} \hat{\alpha}_j s_j \geq 1, \ \forall s \in \hat{R}_f \\ & \alpha_j \geq 0 \ \forall j \in M \end{aligned}$$

From Figure 1, we see that $\hat{\alpha}_1 = \hat{\alpha}_2 = 2$ *and* $\hat{\alpha}_3 = \frac{2}{3}$ *is a feasible solution. Let us prove that this cut is the optimal solution.*
First note that the objective value of the cut is $5.\bar{1}$ *and that* $\left(\frac{1}{4}, 0, \frac{3}{4}\right)$, $\left(0, \frac{1}{4}, \frac{3}{4}\right)$, $\left(\frac{1}{8}, \frac{3}{8}, 0\right)$ *and* $\left(\frac{3}{8}, \frac{1}{8}, 0\right)$ *belongs to* $\hat{R}_f$.
This implies that any cut must satisfy $\hat{\alpha}_1 + \hat{\alpha}_2 \geq 4$ *and* $\hat{\alpha}_1 + \hat{\alpha}_2 + 6\hat{\alpha}_3 \geq 8$.

Let $\hat{Z}$ be the objective value of the optimal solution, then for the optimal cut we have

$$\hat{Z} = \hat{\alpha}_1 + \hat{\alpha}_2 + \frac{5}{3}\hat{\alpha}_3 \geq 8 - \frac{13}{3}\hat{\alpha}_3.$$

Therefore, $\frac{5}{3}\hat{\alpha}_3 \geq \frac{5}{13}(8-\hat{Z})$. Now,

$$\begin{aligned}\hat{Z} &= \hat{\alpha}_1 + \hat{\alpha}_2 + \frac{5}{3}\hat{\alpha}_3 \\ &\geq 4 + \frac{5}{13}(8-\hat{Z})\end{aligned}$$

This inequality implies that $\hat{Z} \geq 5.\bar{1}$, hence our cut is optimal.
This cut, in the original problem corresponds to $\alpha = T^t\hat{\alpha}$, explicitly $\alpha_1 = \alpha_2 = 2$, $\alpha_3 = \frac{2}{3}$ and $\alpha_4 = \frac{4}{9}$. As the objective value is $5.\bar{1}$ which is greater than 4.72, we conclude that solving the reduced problem does not always gives us the optimal solution.

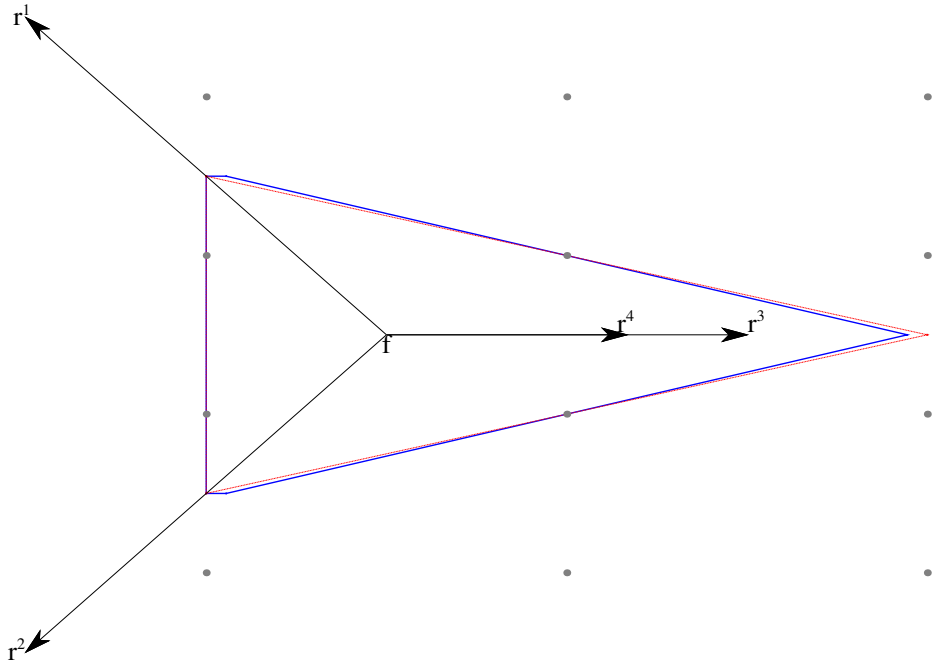

Figure 1: The polyhedron in red corresponds to L_α, while the blue one corresponds to $L_{\hat{\alpha}}$. Both are valid cuts, since there are no integer points in their interior.

3.3 Approximation

The number of iterations of the column generation process is very sensitive to the number of rays. The idea of the approximation is to consider a fixed set of rays and try to deduce a cut using those. Also, it can be seen as a natural question that arises from previous research. In the introduction, we mentioned that in some computational experiments the lattice free set used to separate was fixed, so a natural follow up is to fix the rays.
Let $m > 0$ be an integer number and consider a set of rays $\left\{q^1, \ldots, q^m\right\}$ such that cone $\left(q^1, \ldots, q^m\right) = \mathbb{R}^q$.
We will decompose each ray r^j in terms of the new rays. Let $j \in N$, then there exists $\left\{\theta_{ij}\right\}_{i=1}^m \subseteq \mathbb{R}_+$ such that

$$r^j = \sum_{i=1}^m \theta_{ij} q^i.$$

Define T by $T(s) = \hat{s} \in \mathbb{R}^m$ where $\hat{s}_i = \sum_{j \in N} \theta_{ij} s_j$ and $\hat{R}_f = R_f\left(q^1, \ldots q^m; T(u); \mathscr{S}\right)$.

Proposition 3.8. *Let T be defined as above, then T is a transformation from R_f to $\hat{R}_f$.*

Proof. Clearly, T is linear. We just have to prove that if $s \in R_f$, then $T(s) \in \hat{R}_f$.
Let $s \in R_f$, $\hat{s} = T(s)$ and $\hat{u} = T(u)$. We have to check that $f + \sum_{i=1}^m \hat{s}_i q^i \in \mathscr{S}$ and that $\hat{s}_i \leq \hat{u}_i$.
Indeed,

$$\begin{aligned} f + \sum_{i=1}^m \hat{s}_i q^i &= f + \sum_{i=1}^m \left(\sum_{j \in N} \theta_{ij} s_j \right) q^i \\ &= f + \sum_{j \in N} s_j \left(\sum_{i=1}^m \theta_{ij} q^i \right) \\ &= f + \sum_{j \in N} s_j r^j \in \mathscr{S} \end{aligned}$$

and

$$\begin{aligned} \hat{s}_i &= \sum_{j \in N} \theta_{ij} s_j \\ &\leq \sum_{j \in N} \theta_{ij} u_j = \hat{u}_i \end{aligned}$$

□

Let us make explicit the Γ matrix that appears when solving the transformed problem (8). We have that $T(s) = \Theta s$ where $\Theta_{ij} = \theta_{ij}$, then the objective function of (8) is $||T^t \hat{\alpha}|| = ||\Theta^t \hat{\alpha}||$ and

$$\left(\Theta^t \hat{\alpha}\right)_j = \sum_{i=1}^{m} \theta_{ij} \hat{\alpha}_i$$

In the ℓ^1-norm case, we can write the objective as follows.

$$\begin{aligned} \left|\left|T^t \hat{\alpha}\right|\right|_1 &= \sum_{j \in N} \left(\Theta^t \hat{\alpha}\right)_j \\ &= \sum_{j \in N} \sum_{i=1}^{m} \theta_{ij} \hat{\alpha}_i \\ &= \sum_{i=1}^{m} \left(\sum_{j \in N} \theta_{ij} \right) \hat{\alpha}_i \end{aligned}$$

Then, $\Gamma_{ii} = \sum_{j \in N} \theta_{ij}$.
The ℓ^2-norm case is special since the objective function has cross products and, therefore, there is no diagonal matrix Γ such that $||T^t \hat{\alpha}||_2 = ||\Gamma \hat{\alpha}||_2$.
Trying to solve (8) with ℓ^2-norm would introduce n variables in the dual, while one of the objectives of approximation was to reduce the number of variables. Instead, we change the objective function from

$$\sqrt{\sum_{j \in N} \left(\sum_{i=1}^{m} \theta_{ij} \hat{\alpha}_i \right)^2}$$

to

$$\sqrt{m \sum_{i=1}^{m} \left(\sum_{j \in N} \theta_{ij} \right)^2 \hat{\alpha}_i^2} \tag{10}$$

This allows us to have a smaller dual since now there exists a diagonal matrix Γ, that is given by $\Gamma_{ii} = \sum_{j \in N} \theta_{ij}$ as in the ℓ^1-norm case.
Let us show that not much is lost when applying this change.

Proposition 3.9. *If the objective function of (8) is changed to (10) a m-approximation is obtained.*

Proof. This follows from the equivalence between ℓ^1-norm and l^2-norm, i.e., from the inequality

$$||x||_2 \leq ||x||_1 \leq \sqrt{m}||x||_2, \quad \forall x \in \mathbb{R}^m.$$

Let Z be the optimal value of (8) and $\hat{Z}$ the optimal value of the same problem, but with the new objective function. We have to prove that $Z \leq \hat{Z} \leq mZ$. Indeed,

$$\begin{aligned}
\left|\left|T^t\hat{\alpha}\right|\right|_2 &= \left|\left|\left(\sum_{i=1}^{m}\theta_{ij}\hat{\alpha}_i\right)_{j\in N}\right|\right|_2 \\
&\leq \left|\left|\left(\sum_{i=1}^{m}\theta_{ij}\hat{\alpha}_i\right)_{j\in N}\right|\right|_1 \\
&= \sum_{i=1}^{m}\left(\hat{\alpha}_i\sum_{j\in N}\theta_{ij}\right) \\
&= \left|\left|\left(\hat{\alpha}_i\sum_{j\in N}\theta_{ij}\right)_{i=1}^{m}\right|\right|_1 \\
&\leq \sqrt{m}\left|\left|\left(\hat{\alpha}_i\sum_{j\in N}\theta_{ij}\right)_{i=1}^{m}\right|\right|_2
\end{aligned}$$

Thus, we have that $Z \leq \hat{Z}$.
For the other inequality we have to retrace our steps:

$$\begin{aligned}
\sqrt{m}\left|\left|\left(\hat{\alpha}_i\sum_{j\in N}\theta_{ij}\right)_{i=1}^{m}\right|\right|_2 &\leq \sqrt{m}\left|\left|\left(\hat{\alpha}_i\sum_{j\in N}\theta_{ij}\right)_{i=1}^{m}\right|\right|_1 \\
&= \sqrt{m}\left|\left|\left(\hat{\alpha}_i\sum_{j\in N}\theta_{ij}\right)_{i=1}^{m}\right|\right|_1 \\
&= \sqrt{m}\left|\left|\left(\sum_{i=1}^{m}\theta_{ij}\hat{\alpha}_i\right)_{j\in N}\right|\right|_1 \\
&\leq m\left|\left|\left(\sum_{i=1}^{m}\theta_{ij}\hat{\alpha}_i\right)_{j\in N}\right|\right|_2 \\
&= m\left|\left|T^t\hat{\alpha}\right|\right|_2
\end{aligned}$$

Hence, $\hat{Z} \leq mZ$ $\square$

Still the set of rays, $\{q^1, \ldots, q^m\}$ is undefined. We choose two sets of rays. The first one consists of all the canonical vectors plus the negative of the canonical vectors and complete diagonals, i.e., $(\delta_1, \ldots, \delta_q)$ where $\delta_i \in \{-1, 1\}$ for $i = 1, \ldots, q$. This is called the type 1 approximation.
The second set of rays, called the type 2 approximation, is

$$\{(\delta_1, \ldots, \delta_q) : \delta_i \in \{-1, 0, 1\} \text{ for } i = 1, \ldots q\} \setminus \{0\}.$$

Notice that both sets are the same when $q = 2$.
For the first set we have that $m = 2q + 2^q$ and for the second set, $m = 3^q - 1$.
The idea of the first set is that it is the simplest non trivial one (the trivial set is the set of canonical vectors with their negatives). We consider the second set to determine if more rays give us better cuts. Heuristically, this makes sense, since more rays means that we have more degrees of freedom to approximate the convex set related with the cut. Obviously, this are two of many possible approximations. In the next example we show that this approximation scheme can be arbitrarily bad.

Example 3.10. *Consider the rays* $r^1 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$, $r^2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$, $r^3 = \begin{pmatrix} n+1 \\ n \end{pmatrix}$ *and* $f = \frac{1}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$, *and let* $R_f = R_f(r^1, r^2, r^3; \infty; \mathbb{Z}^2)$. *The rays of* $\hat{R}_f$ *are* $q^1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $q^2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $q^3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $q^4 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$, $q^5 = -q^1$, $q^6 = -q^2$, $q^7 = -q^3$ *and* $q^8 = -q^4$ *(see Figure 2).*

Let Z be the optimal value of the original problem and $\hat{Z}$ *the optimal value of the approximated problem.*
We have the following decomposition of r rays into q rays:

$$r^1 = q^6$$
$$r^2 = q^4$$
$$r^3 = nq^2 + q^3$$

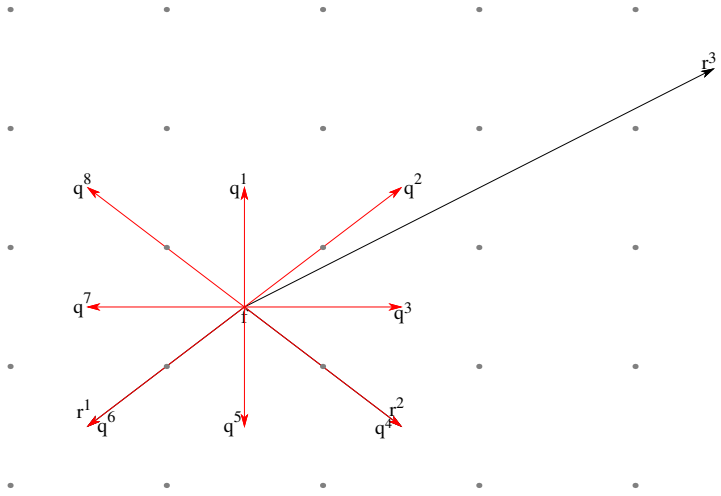

Figure 2: The black rays correspond to R_f rays and the red ones to $\hat{R}_f$ rays. The dots are the integer lattice and $n = 2$.

Then, the approximated problem is

$$\begin{aligned} \min \quad & ||(\hat{\alpha}_6, \hat{\alpha}_4, n\hat{\alpha}_2 + \hat{\alpha}_3)|| \\ s.t \quad & \\ & \hat{\alpha} \cdot \hat{s} \geq 1, \ \forall \hat{s} \in \hat{R}_f \\ & \hat{\alpha}_i \geq 0, \ \text{for } i = 1, \ldots, m \end{aligned}$$

Clearly $\hat{s} = \left(0, \frac{1}{2}, 0, 0, 0, 0, 0, 0\right) \in \hat{R}_f$*, therefore the constraint* $\frac{1}{2}\hat{\alpha}_2 \geq 1$ *belongs to the approximated problem and in the optimal solution* $\hat{\alpha}_2 \geq 2$*. It follows that* $\hat{Z} \geq 2n$*.*
On the other hand, $\alpha_1 = \alpha_2 = \alpha_3 = 2$ *is a feasible point for the original problem (see Figure 3). To see this we will show that there is no* $\begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{Z}^2$ *such that* $\begin{pmatrix} x \\ y \end{pmatrix} = f + r^1 s_1 + r^2 s_2 + r^3 s_3$ *and* $s_1 + s_2 + s_3 < \frac{1}{2}$*.*
If such point should exist, then $x - y = 2s_2 + s_3 \in \mathbb{Z}$*. Recall that* $s_1, s_2, s_3 \geq 0$ *and since* $s_1 + s_2 + s_3 < \frac{1}{2}$ *we have that* $s_1, s_2, s_3 \leq \frac{1}{2}$*, so* $0 \leq 2s_2 + s_3 \leq \frac{3}{2}$*. The integrality of* $x - y$ *implies that* $2s_2 + s_3 \in \{0, 1\}$*.*
If $2s_2 + s_3 = 0$*, then* $s_2 = s_3 = 0$ *and* $s_1 \geq \frac{1}{2}$ *contradicting the hypothesis.*
If $2s_2 + s_3 = 1$*, then* $2s_1 + 2s_2 + 2s_3 \geq 2s_2 + s_1 = 1$*, therefore* $s_1 + s_2 + s_3 \geq \frac{1}{2}$ *also*

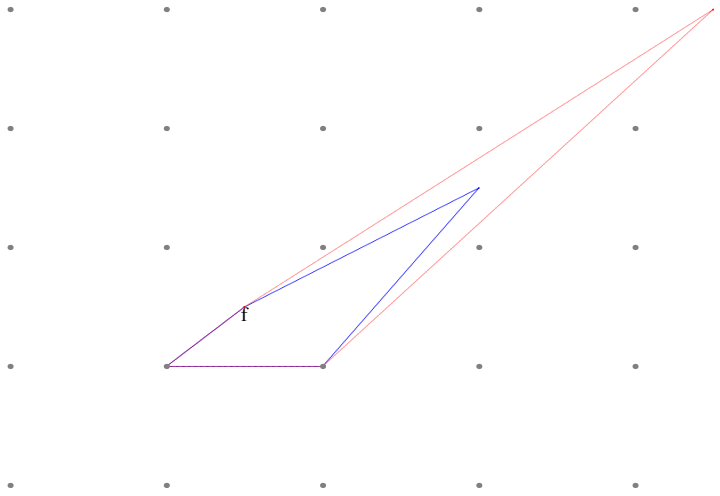

Figure 3: Here we appreciate the set L_α for $\alpha = (2,2,2)$ for $n = 2$ (the blue one) and $n = 5$ (the red one). Notice that both are lattice free, hence the cut is valid.

contradicting the hypothesis.
This shows that $\alpha_1 = \alpha_2 = \alpha_3 = 2$ *is a feasible point, thus* $Z \leq 6$.
We see then that the approximated solution can be arbitrarily bad.

Even though this example shows that there is no warranty on the quality of the solution provided by the approximation, we will show in the results that approximated cuts can be useful.
Let us conclude this section with a final advantage of this approximation. As the rays are fixed, the only thing that changes is the f vector, therefore we can use the s points we have already computed, translating them so that they become valid points for the new f.

4 Building the relaxation

In this section we will show how to build the relaxation, i.e., the X set. In general, the procedure is as follows.

1. Solve the LP relaxation.
2. Compute the tableau of basic integer variables.
3. Select rows.
4. Set Γ as the identity, apply options and transformations.

In the previous section we saw the transformations and how to use them, as well as the Γ matrices involving each of them. Now, we will discuss how to select rows and the rest of the options.

4.1 Row selection

We have to select q rows from the tableau. The criteria for row selection that we use are the following.

Random We select q random rows. We need at least one component of f to be fractional, so the first row is selected randomly over all rows that have a fractional right hand side and the $q-1$ left are selected over all rows.

Biggest dot product We select one row randomly over the rows that have a fractional right hand side. Then we compute the dot product between the selected row and all others and choose the $q-1$ that has the biggest dot product.

Smallest dot product The same as before, but now the smallest dot product is selected instead of biggest.

The idea is to see if selecting rows using the biggest or smallest dot product is better than random. There are many other rows selection criteria that can be tested, however we focus on these two criteria, basically, because of their simplicity.

4.2 Bounds

We consider the 4 scenarios: Bounds on both the x and s variables, only on x, only on s and bounds on none of them.
This is a central contribution of our work, since, to the best of our knowledge, no tests have been made with bounds on both variables.

4.3 Modification of non-basic integer variables

Here we take advantage of the integrality of some nonbasic variables and modify the rays associated to them. The modification used is one of many possible modifications. We try that both rays, the original and the modified, point towards the same orthant. Specifically, let r^j be a ray associated with an integral nonbasic variable and let $[x]$ be the rounding towards 0 function, i.e.,

$$[x] = \begin{cases} \max\{n \in \mathbb{Z} \colon n \leq x\}, & x \geq 0 \\ \min\{n \in \mathbb{Z} \colon n \geq x\}, & x \leq 0 \end{cases} \tag{11}$$

Also, let $\{x\} = x - [x]$, then for each $j \in N$ that corresponds to an integer variable, we transform r^j into $\hat{r}^j$ defined by

$$\hat{r}_i^j = \left\{ r_i^j \right\}, \quad i = 1, \ldots, q$$

The remaining part, $\left(r^j - \hat{r}^j\right) s_j$, is absorbed by the x variables.
When we are asked to separate using bounds on x variables, bounds on s will be computed and used to calculate the new bounds on x due to the addition of $\left(r^j - \hat{r}^j\right) s_j$.
This modification is particularly useful when the ray to be modified is integral, because the remaining part is the zero vector and the number of rays is reduced.
Another heuristic argument in favor of this modification is that by reducing the norm of the ray, one would expect to obtain a cut with smaller coefficients. This may help the numerical stability of the cuts and may produce deeper cuts.

5 Numerical issues and algorithmic improvements

In this section we first show how we can use some optimality conditions to improve the performance of the overall process. Then we discuss how we deal with numerical issues.

5.1 Optimality conditions and algorithmic improvements

Assume that $R_f = R_f\left(r^1,\dots,r^n;\infty;Z^q\right)$, i.e., the s variables do not have upper bounds and the x variables are free. We can use the optimality condition presented in Proposition 2.6 to improve the performance of the column generation.
During the column generation, we can ask for the v^j that are not at the boundary and introduce a new constraint in (2) (variable in the master) that forces v^j to be at the border. Tests done using this new constraints show that the time spent searching for the cut can be reduced.
Below, we deduce this constraints. For this we need to work with the translation of L_α by f,

$$K = L_\alpha - f = \operatorname{conv}\left(\{0\} \cup \left\{\frac{r^j}{\alpha_j}\right\}_{j\in N_+}\right) + \operatorname{cone}\left(\left\{r^j\right\}_{j\in N\setminus N_+}\right).$$

Definition 5.1. *The **guage function** ψ is defined by*

$$\psi(x) = \inf\left\{\lambda > 0 \colon \frac{x}{\lambda} \in K\right\}$$

It is useful to study the gauge function since the boundary of K (denoted by ∂K) satisfies $\partial K = \{x \in \mathbb{R}^q \colon \psi(x) = 1\}$.
The *polar* of K (see [22] for standard definitions and notation) is given by

$$K^\circ = \left\{z \in \mathbb{R}^q \colon z \cdot r^j \le \alpha_j, \text{ for } j \in N\right\}$$

We will use the following theorem proved in [21].

Theorem 5.2. *Let C be a closed convex set containing the origin. The polar C° is another closed convex set containing the origin. The gauge function ψ satisfies*

$$\psi(x) = \max\{x \cdot z \colon z \in C^\circ\} \tag{12}$$

Note that $0 \in K$ and that K is a polyhedron, hence it is a closed convex set. Also, for $j \in N_+$, $\frac{r^j}{\alpha_j} \in \partial K$ if and only if $\psi\left(\frac{r^j}{\alpha_j}\right) = 1$. The last equality is equivalent to $\psi\left(r^j\right) = \alpha_j$.

Definition 5.3. *Let α be a cut candidate during the column generation process. We say that α violates a* ***boundary constraint at*** *i if $i \in N_+$ and $\psi\left(r^i\right) < \alpha_i$.*

Proposition 5.4. *Let α be a cut candidate during the column generation process and $i \in N_+$. If α violates a boundary constraint at i, then there exist $\lambda_1, \ldots, \lambda_N \in \mathbb{R}_+$ such that*

$$\begin{aligned} \lambda_i &= 0, \\ \sum_{j \in N \setminus \{i\}} \lambda_j r^j &= r^i \text{ and} \\ \sum_{j \in N \setminus \{i\}} \lambda_j \alpha_j &< \alpha_i \end{aligned} \tag{13}$$

Proof. The result follows from the dual of (12), i.e., the dual of $\psi\left(r^i\right)$,

$$\begin{aligned} \max \quad & z \cdot r^i \\ \text{s.t.} \quad & \\ & z \cdot r^j \leq \alpha_j \; \forall j \in N \end{aligned}$$

which is

$$\begin{aligned} \min \quad & \sum_{j \in N} \lambda_j \alpha_j \\ \text{s.t.} \quad & \\ & \sum_{j \in N} \lambda_j r^j = r^i \\ & \lambda_j \geq 0 \end{aligned} \tag{14}$$

From duality theory and $\psi\left(r^i\right) < \alpha_i$, we have that the optimal solution λ^* of (14) satisfies $\sum_{j \in N} \lambda_j^* r^j = r^i$ and $\sum_{j \in N} \lambda_j^* \alpha_j < \alpha_i$.
If $\lambda_i^* > 0$, then necessarily $\lambda_i^* < 1$ and by setting $\lambda_j = \frac{\lambda_j^*}{1-\lambda_j^*}$ for $j \neq i$ and $\lambda_i = 0$, we have the desired result.
If $\lambda_i^* = 0$, set $\lambda_j = \lambda_j^*$ for all $j \in N$ to conclude. □

Now it is clear that the sought constraints are of the form

$$\sum_{j \in N \setminus \{i\}} \lambda_j \alpha_j \geq \alpha_i,$$

where the λ vector is obtained by solving (14).
The problem with this constraints are that they only apply for the unbounded case.

Actually, for the general case, Proposition 2.6 does not hold, so we need a new optimality condition. We use the following trivial condition: if α solves (2), then it must be feasible.
Let $R_f = R_f\left(r^1, \ldots, r^n; u; \mathscr{S}\right)$.

Definition 5.5. *Let α be a cut candidate during the column generation process and $x \in \mathscr{S}$. We will say that x has a **violated representation** if there exists $s \in R_f$ such that $\alpha \cdot s < 1$.*

In our column generation process, we can use this to search for violated representations of points that we already now, instead of looking for any (x,s) such that s violates the cut. Tests done using this simple idea, proves to speed up the process as well.
We implemented this as follows:

1. Solve the master problem and obtain the cut candidate α
2. For every basic β_s of the master, check if $x = f + \sum_{j \in N} s_j r^j$ has a violated representation.
3. Select the nearest x to f that has a violated representation. If it does not exist, solve the subproblem.
4. Add the s that represents x to the master and repeat.

Basically, instead of solving (5), we fix the value of the x variables and solve the remaining LP.
Apart from being applicable in any R_f, this procedure generalizes the previous one in the sense that any time there is a violation to a boundary constraint, we can find a violated representation (only on the cases where boundary constraint apply, i.e., in the unbounded case).
To prove this, let $R_f = R_f\left(r^1, \ldots, r^N; \infty; Z^q\right)$.
Suppose we are at some point in our column generation process and recall that R'_f is the set of s points that we have found so far. For each $s \in R'_f$ let $x_s = f + \sum_{j \in N} s_j r^j$.
We will now show that if there is any $i \in N_+$ such that α has a violated boundary constraint at i, then there is a x_s that has a violated representation.

Proposition 5.6. *Let α be a cut candidate during the column generation process and $i \in N_+$ such that α violates a boundary constraint at i. Then there is a basic variable β_σ such that x_σ has a violated representation.*

Proof. Let us find σ. Since $i \in N_+$, $\alpha_i > 0$ and by complementary slackness (CS) we have that $\sum_{s \in R'_f} \beta_s s_i = 1$. This equality implies that there exists a $\sigma \in R'_f$ such that $\beta_\sigma > 0$ and $\sigma_i > 0$.
Now by CS we have that $\sum_{j \in N} \alpha_j \sigma_j = 1$. By Proposition 5.4 there exists $\lambda_1, \ldots, \lambda_N \in \mathbb{R}_+$ that satisfies (13).
We are now ready to give the violated representation of x_σ.

$$\begin{aligned} x_\sigma &= f + \sum_{j \in N} \sigma_j r^j \\ &= f + \sum_{j \in N \setminus \{i\}} \sigma_j r^j + \sigma_i r^i \\ &= f + \sum_{j \in N \setminus \{i\}} \sigma_j r^j + \sigma_i \sum_{j \in N \setminus \{i\}} \lambda_j r^j \\ &= f + \sum_{j \in N \setminus \{i\}} \left(\sigma_j + \sigma_i \lambda_j\right) r^j \end{aligned}$$

Finally,

$$\begin{aligned} \sum_{j \in N \setminus \{i\}} \left(\sigma_j + \sigma_i \lambda_j\right) \alpha_j &= \sum_{j \in N \setminus \{i\}} \sigma_j \alpha_j + \sigma_i \sum_{j \in N \setminus \{i\}} \lambda_j \alpha_j \\ &< \sum_{j \in N \setminus \{i\}} \sigma_j \alpha_j + \sigma_i \alpha_i \\ &= \sum_{j \in N} \sigma_j \alpha_j \\ &= 1 \end{aligned}$$

□

5.2 Numerical issues

Here we explain how we numerically manage the cut candidates, the *s* points, parallelism and tableau's coefficients.

5.2.1 Cut candidates

Every time a cut candidate is obtained, we first check its coefficients. If any value is below the *absolute zero tolerance*, whose value is 10^{-8}, we set it to 0. Then, if any value π_j is below the *zero tolerance*, from now on denoted by ε, whose

default value is 10^{-6}, we ask if there is a known s point such that $\pi_j \cdot s_j > \varepsilon$. If true, then we keep the value of π_j, but if not, we set π_j to 0.
All nonzero coefficients are amplified by the *security tolerance*, whose default value is 1.001. We do this in order to help the column generation process. It is known that, in general, the convergence of column generation is usually slow (see [5] and [16]), so this amplification will stop the process before optimality.
To assure numerical stability, we check that for all known points, $\sum_{j \in N} \pi_j s_j \geq 1$. If for any point this is not true, the separation is aborted. Amplifying by the security tolerance also helps the numerical stability of the process, since its less likely to cut off an integer feasible point.
Let us call (ζ, ζ_0) the cut obtained after we transformed it into the original variables (ζ_0 is the right hand side of the cut). Define its **quality** as $\frac{\zeta_0 - \zeta \cdot x^*}{||\zeta||_2}$ and its **stability** as $\frac{\zeta_{\max}}{\zeta_{\min}}$ where $\zeta_{\max}$ is the biggest coefficient in absolute value, $\zeta_{\min}$ is the smallest nonzero coefficient in absolute value and $x*$ is the optimal solution of the LP relaxation of the MIP that we are trying to solve. We only add the cut if its quality is greater than 10^{-3} and its stability is less than 10^6.

5.2.2 The s points

Many numerical issues arise because of s points. We can obtain numerically unstable master problems if some coefficients of s are large. To try to avoid this, we obtain the first point by using a fake cut candidate with $\pi_j = 1$ for all $j \in N$. For the following points, we use the previous optimal solution of the subproblem as a starting point, hoping that coefficients stay bounded.
Every time a new s is generated, we check for coefficients smaller than ε. If $s_j < \varepsilon$ for some $j \in N$, we set s_j to 0.

5.2.3 Tableau's coefficients

The process starts by solving the LP relaxation of a MIP using SIMPLEX. Afterwords we need to build the optimal tableau. When doing so, we have to decide whether a variable obtains an integral value or not. We say that a value is fractional, when the distance between the value and the nearest integer is greater than 10^{-3}.
However, when we are modifying rays associated to integral nonbasic variables, we consider a ray's coefficient as fractional when the distance between the coefficient and the nearest integer is greater than ε.

The reason for this, is that we want to make sure that $f \notin \mathbb{Z}^q$, i.e., we want f to have a sufficiently fractional component, but when we set to 0 a ray's coefficient, we want to be sure that we are not making a gross approximation.
For the coefficients of the tableau associated to the basic variables (which should form a permutation of the identity matrix), we check that the 0 and 1 coefficients are really 0 and 1, by asking if the distance between the coefficient and 0 is less than ε and that the distance between the coefficient and 1 is less than ε respectively. If any of these coefficients fail to be 0 or 1 enough, we abort the construction of the tableau due to numerical instability.
We treat rays' small coefficients (coefficients of the tableau associated to the non-basic variables) similarly as cut's coefficients. Let r^j be a ray with $j \in N$. When we find a small coefficient, i.e., $r^j_i \leq \varepsilon$, we ask if the maximum value achievable by s_j can make the product $r^j_i s_j$ greater than ε. If yes, we leave the coefficient as it is and if no, we set it to 0. Finally, given two rays r^1 and r^2, we regard them as parallel if there exists $\lambda > 0$ such that $\left|\left|\lambda r^1 - r^2\right|\right|_\infty \leq \varepsilon$.

6 Experiments & Results

6.1 The Balas experiment

Our first experiment consist on computing all possible cuts deduced from exactly q rows from the tableau obtained by solving the LP relaxation. In other words, we read the problem, solve the LP relaxation and obtain the tableau. Then, suppose we are separating with $q = 3$ and that our tableau has 5 rows. We deduce cuts with rows 1-2-3 of the tableau, then 1-2-4, 1-2-5, and so on.
We compare these results against Balas and Saxena results on the split closure (see [4]), so our test bed is a subset from MIPLIB3. We ran our experiments with a time limit of 2 hours.
For all the runs, we removed parallel rays and we limited the number of iterations to 10000. Also we use CPLEX basic presolve, which only tightens the bounds. We *did not* use the approximation scheme. When we minimize the ℓ^2-norm, we use a 0.01-approximation of the norm.
We remove problems whose LP value and optimal value are the same as their optimal value (dsbmip, enigma and noswot), the ones that did not finish (air04, air05, fast0507, mitre, seymour, 10teams, dano3mip, danoint, mod11 and qnet1) and problems for which ours and Balas procedure were not able to improve the LP value (stein27, stein45, pk1).
The percentage of gap closed is computed as

$$100 - 100 \times \frac{\text{remaining gap}}{\text{original gap}}$$

In Table 6.1 we present the results of the configuration that managed to finish more instances against Balas. The configuration consists of: 2 rows, bounds only on x variables, ℓ^1-norm and the modification of nonbasic integer variables.

Now we can compare the effect of each option over the closed gap against our base configuration. For this, we will compute the geometric average of the gap closed only on the instances that both configurations managed to solve. We only considered positive gaps.
When we do not modify nonbasic integer variables, we obtained 38 problems in common with our base configuration. We obtained 22.87% of closed gap when we modified and 18.17 % of closed gap when we did not. When no bounds were consider, there were 36 problems in common and the closed gaps were 22.42% and 23.45% for no bounds and bounds on x variables, repectively. If we considered bounds on both variables we had 35 problems in common and we got a closed gap

Instance	gap	Balas gap	Instance	gap	Balas gap	Instance	gap	Balas gap
p0282	4.14	99.90	air03	99.90	100.00	gesa3	69.99	95.78
p0548	50.36	100.00	cap6000	42.38	37.63	gesa3_o	71.58	95.31
p2756	0.72	92.32	fiber	13.70	98.50	khb05250	74.81	100.00
arki001	21.75	83.05	gt2	57.60	100.00	misc06	41.29	100.00
bell3a	44.66	55.19	harp2	9.10	17.50	modglob	21.02	96.48
bell5	14.51	87.44	l152lav	5.93	92.10	pp08aCUTS	34.02	97.01
blend2	16.41	46.77	lseu	33.62	93.75	qiu	2.57	77.51
dcmulti	42.37	100.00	misc03	8.61	51.47	qnet1_o	43.99	100.00
egout	96.58	100.00	misc07	0.72	19.48	rgn	8.98	100.00
fixnet6	31.75	99.76	mod008	1.40	100.00	rout	2.70	70.73
flugpl	12.43	100.00	mod010	97.57	100.00	set1ch	59.37	89.41
gen	49.51	100.00	nw04	66.00	100.00	vpm1	11.28	100.00
gesa2	33.61	98.66	p0033	10.49	87.42	vpm2	21.56	81.22
gesa2_o	35.65	100.00	p0201	33.75	74.93			

Table 1: Comparison between multi-row cuts generated from every tableau with 2 rows with bounds on x, ℓ^1-norm and modification of nonbasic integer variables, and Balas split closure.

of 23.33%, against a 23.06% for bounds only on the x variables. When comparing the norms, we had 38 problems in common and we found that ℓ^1-norm performed better than ℓ^2-norm, closing 22.08% of gap against 21.48%. Finally, if we constructed tableau with 3 rows, we had 24 problems in common and 18.06% of the gap was closed, while only a 16.25% of the gap was closed with 2 row tableau.

6.2 The locally optimal configuration

For the following experiments, our procedure was called as a cut callback on CPLEX and the maximum number of iterations was limited to $5 \times$ Number of rays. Calling Z the value obtained by our procedure, the gap was computed as

$$100 - 100 \times \frac{\text{MIP}_{\text{OPT}} - Z}{\max\{1, |\text{MIP}_{\text{OPT}}|, \text{MIP}_{\text{OPT}} - \text{LP}_{\text{OPT}}\}}$$

Using this formula, we ensure that the gap is between 0 and 100 and that we do not divide by 0.

Our test bed was the benchmark subset of MIPLIB 2010, where we eliminated all the problems for which some configuration could not add any cut. These problems were ex9 and mspp16.

We decided that our procedure would only add cuts at the root node and every

time the cut callback was called, it tried to add 20 cuts. If 1000 cuts had been added, then no more cuts were generated. Notice that the cut callback can be called several times on the root node.
On our first experiment, we ran with a time limit of 2 hours and we stopped the optimization when the root node was solved.
We have that the configuration: 4 rows, bounds on all variables, type 2 approximation, modification of nonbasic integer variables, security tolerance of 1.001, select rows using the least dot product criteria and minimization of the ℓ^1-norm, is locally optimal in the sense that if we changed any single parameter while keeping the others, the results were worse. We used MIPLIB2003 to obtain a starting point to search for this locally optimal configuration.
In Table 2, we associate numbers to the configurations considered, and in Figure 4 we show the results. The percentage of closed gap is the geometric average of the closed gaps and the time factor is the geometric average of the time spent for each instance normalized by the best time of that instance across all configurations. We show the results on logarithmic scale.

Number	Configuration	Number	Configuration
0	r4 a2 M1 b0 m2 T1.001 N0	9	r2 a2 M1 b3 m2 T1.001 N0
1	r5 a2 M1 b3 m2.T1.001 N0	10	r4 a2 M1 b3 m2 T1.1 N0
2	r3 a2 M1 b3 m2 T1.001 N0	11	r4 a2 M1 b3 m1 T1.001 N0
3	r4 a2 M1 b3 m2 T1.001 N0	12	r4 a2 M0 b3 m2 T1.001 N0
4	r4 a2 M1 b3 m2 T1.001 N1	13	r4 a2 M1 b3 m2 T1.0001 N0
5	CPLEX	14	r4 a2 M1 b3 m2 T1.01 N0
6	r4 a1 M1 b3 m2 T1.001 N0	15	r4 a2 M1 b1 m2 T1.001 N0
7	r4 a2 M1 b3 m2 T1.00001 N0	16	r4 a2 M1 b3 m0 T1.001 N0
8	r4 a0 M1 b3 m2 T1.001 N0	17	r4 a2 M1 b2 m2 T1.001 N0

Table 2: Configurations: r means the number of rows, a means the type of approximation, M means if we modify nonbasic integer variables, b means on which variables we consider bounds (0 is none, 1 is only s, 2 is only x and 3 is both), m means the method of row selection used (0 is random, 1 biggest dot product and 2 lowest dot product), T means the security tolerance and N means if we use Ben-Tal and Nemirovski approximation (optimize the ℓ^2-norm using a 0.01-approximation of the norm).

Finally, we wanted to see how multi-row cuts affect the branch and bound tree, so we compare our locally optimal configuration against CPLEX, but this time we

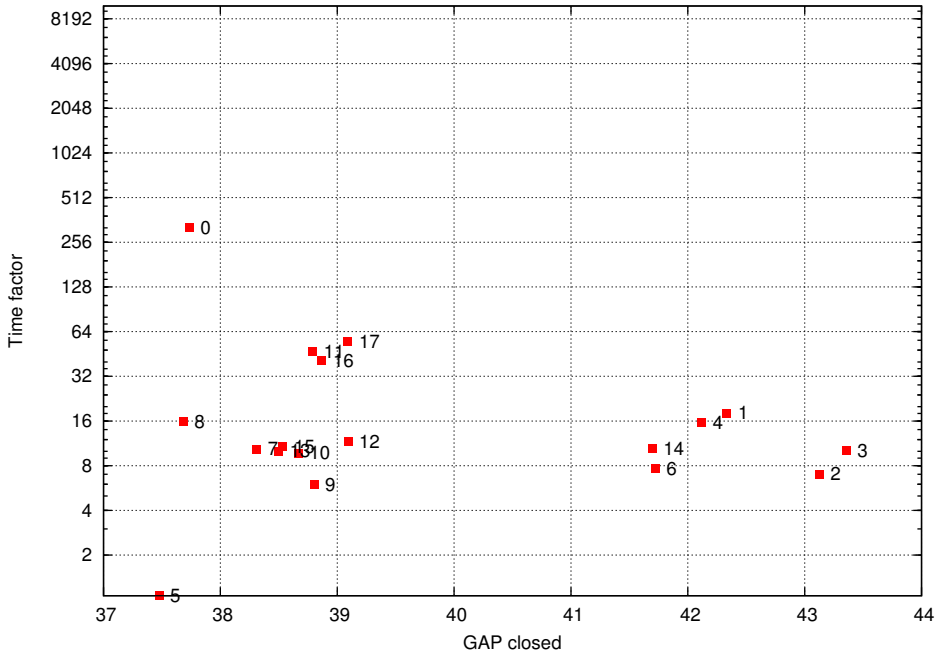

Figure 4: Sensitivity analysis for r4 a2 M1 b3 m2 T1.001 N0 over the benchmark subset of MIPLIB2010.

ran with a 4 hour time limit and we did not stop the optimization once the root was solved. However, cuts were only added at the root node.
The results are shown on Figure 5. The geometric average of the closed gap is given in the following table.

Configuration	% closed GAP (avg)
CPLEX	78.31
OPTCONF	85.09

The difference between the gap averages is explained by the problem ns1208400. The optimal value of this problem is 2, while its LP value is 0. Our procedure managed to obtain a lower bound of 1 while CPLEX could not improve the LP value. If we remove this problem, we obtain the following averages.

Configuration	% closed GAP (avg)
CPLEX	87.13
OPTCONF	85.63

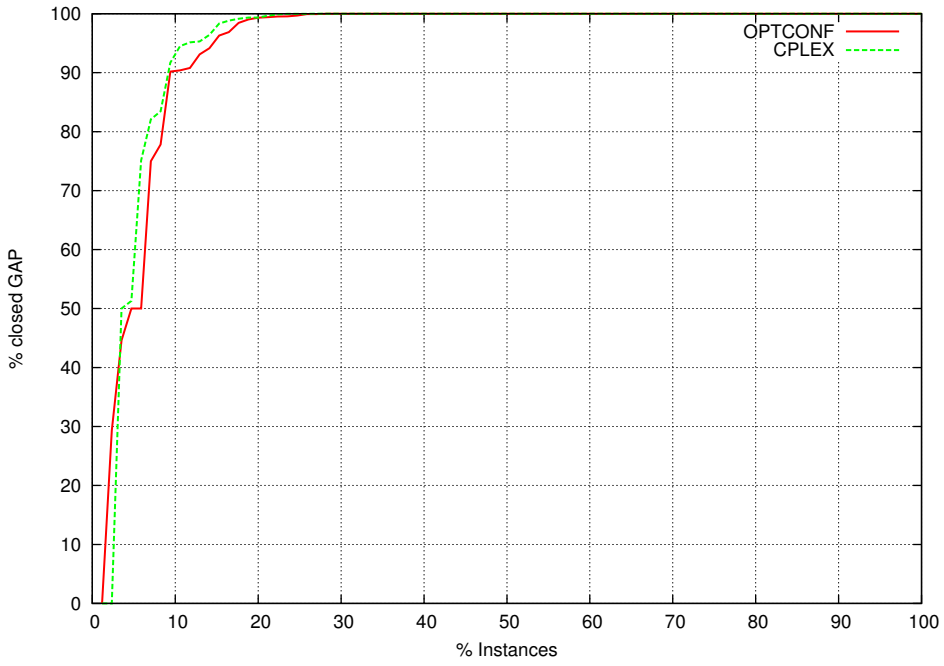

Figure 5: Performance profile: runing over 4 hours CPLEX and our configuration.

7 Conclusions and future work

When multi-row cuts where introduced, Andersen et al. showed an example in which a single cut was needed to solve the problem. Here we observed also some problems in which multi-row cuts worked better than existing techniques. For example, cap6000 in Table 6.1 or ns1208400 as shown in the last section. This suggests that multi-row cuts may be very useful, but it is not yet clear how to generate them. In particular, the question "How to select the rows of the tableau?" seems to be the most important.

Comparing rank 1 cuts, as in the Balas experiment, provides us with the fairest possible scenario for comparing different configurations, since this avoids the feedback inherent to the cutting process (i.e. whenever we add an active cut, the resulting optimal tableau depends on the cuts previously added). Unfortunately, the current implementation of our algorithm does not allow us to explore bigger problems or more than 3 rows.

Regarding the question of how the number of rows affects the cuts generated, our experiments showed that up to 4 rows we get an improvement. However, further tests are required, given that we observed the added effect of multi-row cuts with other types of cuts without trully understanding how they relate.

On a more positive side, it is clear that even a simple modification on nonbasic integer variables generates better cuts. This is clearly shown both in the Balas experiment and in the sensitivity analysis. Also, using l^2-norm does not give better results than l^1-norm, while considering bounds, at least on the x variables, proved to be better than not doing so.

The approximation scheme allowed us to generate cuts on bigger problems and, as results showed, may be a useful way to produce multi-row cuts in a general purpose solver. The reason why approximated cuts performed better than non approximated cuts is that, while approximated configurations solved the root node, the non approximated configuration did not because of the impossed time limit. Tests with more time are needed in order to see how much we lose when we approximate.

It is our belief, that for multi-cuts to be useful we have to understand, first, how to obtain the tableaus and then how do multi-row cuts relate to other kinds of cuts. Regarding the first question, we think that generating cuts deduced from the LP relaxation only, give us a fair framework for comparing the cuts. As for future research, it would be interesting to know how different approximation schemes work. For example, apart from using q rays pointing towards all integer points inside the ball centered at the origin with radius 1, what happens if we consider

the points inside the ball with radius 2. Or, what if we fix a maximum number, N, of rays and form N groups of rays and then we select a representative (that could be the average of the rays) of each group and separate with the representatives. Notice that the results suggest that more fine approximations, i.e., with more rays, may be better.

References

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

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

[3] K. Andersen, Q. Louveaux, R. Weismantel, and L.A. Wolsey. Inequalities from two row of a simplex tableau. *LNCS*, 4513:1–15, 2007.

[4] Egon Balas and Anureet Saxena. Optimizing over the split closure. *Math. Program.*, 113(2):219–240, 2008.

[5] Cynthia Barnhart, Ellis L. Johnson, George L. Nemhauser, Martin W. P. Savelsbergh, and Pamela H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46:316–329, 1996.

[6] Aharon Ben-Tal and Arkadi Nemirovski. On polyhedral approximations of the second-order cone. *Math. Oper. Res.*, 26:193–205, May 2001.

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

[8] Robert E. Bixby, Mary Fenelon, and Zonghao Gu. Mip: Theory and practice - closing the gap. In *System Modelling and Optimization: Methods, Theory, and Applications*. Kluwer Academic Publishers, 2000.

[9] Valentin Borozan and Grard Cornujols. Minimal valid inequalities for integer constraints, 2007.

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

[11] Gérard Cornuéjols and François Margot. On the facets of mixed integer programs with two integer variables and two constraints. In *Proceedings of the 8th Latin American conference on Theoretical informatics*, LATIN'08, pages 317–328, Berlin, Heidelberg, 2008. Springer-Verlag.

[12] Santanu Dey, Andrea Lodi, Andrea Tramontani, and Laurence Wolsey. Experiments with two row tableau cuts. In Friedrich Eisenbrand and F. Shepherd, editors, *Integer Programming and Combinatorial Optimization*, volume 6080 of *Lecture Notes in Computer Science*, pages 424–437. Springer Berlin / Heidelberg, 2010.

[13] Santanu S. Dey and Laurence A. Wolsey. Lifting integer variables in minimal inequalities corresponding to lattice-free triangles. In *Proceedings of the 13th international conference on Integer programming and combinatorial optimization*, IPCO'08, pages 463–475, Berlin, Heidelberg, 2008. Springer-Verlag.

[14] Daniel Espinoza. Computing with multi-row gomory cuts. In *Proceedings of the 13th international conference on Integer programming and combinatorial optimization*, IPCO'08, pages 214–224, Berlin, Heidelberg, 2008. Springer-Verlag.

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

[16] P. C. Gilmore and R. E. Gomory. A Linear Programming Approach to the Cutting Stock Problem–Part II. *OPERATIONS RESEARCH*, 11(6):863–888, November 1963.

[17] Ralph E. Gomory and Ellis L. Johnson. Some continuous functions related to corner polyhedra. *Mathematical Programming*, 3:23–85, 1972.

[18] Thorsten Koch, Tobias Achterberg, Erling Andersen, Oliver Bastert, Timo Berthold, Robert E. Bixby, Emilie Danna, Gerald Gamrath, Ambros M. Gleixner, Stefan Heinz, Andrea Lodi, Hans Mittelmann, Ted Ralphs, Domenico Salvagnin, Daniel E. Steffy, and Kati Wolter. MIPLIB 2010. *Mathematical Programming Computation*, 3(2):103–163, 2011.

[19] Yanjun Li and Jean-Philippe P. Richard. Cook, kannan and schrijver's example revisited. *Discrete Optimization*, pages 724–734, 2008.

[20] Q. Louveaux and L. Poirrier. An algorithm for the separation of two-row cuts, 2011.

[21] R. T. Rockafeller. *Convex Analysis*. Princeton University Press, MA., 1970.

[22] A. Schrijver. *Theory of linear and integer programming*. Wiley, 1986.