# Computing with multi-row Gomory cuts

Daniel G. Espinoza

*Departamento de Ingeniería Industrial, Universidad de Chile, Av. República 701, Santiago, 837-0439, Chile*

**Abstract**

Recent advances on the understanding of valid inequalities from the infinite group relaxation has opened the possibility of finding a computationally effective extension to GMI cuts. In this paper, we investigate the computational impact of using a subclass of minimally-valid inequalities from this relaxation on a wide set of instances.

*Key words:* mixed integer programming, cutting planes, multiple constraints.
*2000 MSC:* 90C11,
*2000 MSC:* 90C57

## 1. Introduction

The most successful approach to solve general MIP today is branch and cut, where general cutting planes are a crucial factor for the overall performance. After the great success in the 90's of using general purpose cutting planes such as GMI cuts [10, 4], a great deal of research was devoted to extend those ideas to find other families of general cuts that could consistently outperform GMI cuts. However, results have been mixed, and although there are several extensions that in theory are at least as good as GMI cuts, in practice they do not seem to offer much advantage. Most of these extensions have focused on deriving inequalities from the master cyclic group problem introduced by Gomory and Johnson [12], which look at problems with a single linear constraint.

The theoretical importance of looking at multi-row relaxations has been proved in a number of works. Cook et al. [5], show an example with infinite Chvátal-Gomory rank (i.e. obtaining the convex hull of the integer points by adding inequalities derived from one row relaxations is impossible). However, Andersen et al. [2], prove that by looking at inequalities generated from two row relaxations, the convex hull of the Cook-Kannan-Schrijver example, can be obtained by adding a single cut. Yanjun Li and Jean-Philippe P. Richard [15] extend this situation to higher dimension.

An interesting recent development has been the work of Cornuéjols and Borozan [6] and Gomory [11]; who have proposed to look at the so-called infinite

*Email address:* `daespino@dii.uchile.cl` (Daniel G. Espinoza)

relaxation problem, which was also introduced by Gomory and Johnson [12]. A first property of this relaxation, is that it considers several constraints at the same time, thus including cuts as in [2]. Secondly, it focuses on the relation between a few integer variables and many continuous variables at the same time, this may be relevant, since most problems do have integer and continuous variables. Cornuéjols and Borozan [6] show that any minimal valid inequality for the relaxation can be related to maximal, convex, lattice-free polyhedra; thus identifying *relevant* inequalities with simple geometrical entities, moreover Cornuéjols and Margot [7] provide a description of all facets of the infinite (and finite) relaxation with two constraints, while Dey and Wolsey [8] studied the case when some other variables are integer.

To the best of our knowledge, no extensive computational test of the impact of using cuts derived from this relaxation have been published. The main contribution of this paper is to computationally test a wide set of lattice-free sets; show that, at least in terms of root LP integrality gap, there is an advantage at looking at relaxations derived from more than two rows; provide a fairer comparison of the different approaches; and show that some relaxations are also very valuable in practice, not only improving the LP gap closed at the root node, but also in speeding-up the overall branch and cut performance when compared with CPLEX [13] defaults.

The rest of the paper is organized as follows. Section 2 presents the definition and basic results related to the infinite relaxation. Section 3 define the basic ground sets used for generating valid inequalities. Section 4 explain the ideas used to separate each family of inequalities, and selection rules for cut-generation. Section 5 explain our experiments, settings, and results. Finally, Section 6 presents our conclusions and further questions.

Throughout the paper the following notation is used: Given $S \subset \mathbb{R}^n$, a full-dimensional closed set, we denote $S^\circ$ its interior and $\partial S := S \setminus S^\circ$ its boundary.

A preliminary version of the present paper was presented at the 13th international IPCO conference, Bertinoro (Italy) May 26-28, 2008 [9].

## 2. The infinite relaxation

Consider a general mixed integer program (MIP)

$$\min \left\{ cx : Ax = b,\, x \in \mathbb{R}^n_+,\, x_i \in \mathbb{Z}\, \forall i \in I \right\}, \tag{1}$$

where $I \subseteq \{1, \ldots, n\}$, $A \in \mathbb{Q}^{m \times n}$ is of full row rank, $c \in \mathbb{Q}^n$, $b \in \mathbb{Q}^m$. Branch and cut algorithms start by solving

$$\min \left\{ cx : Ax = b,\, x \in \mathbb{R}^n_+ \right\}, \tag{2}$$

the LP relaxation of (1), and obtain an optimal basic feasible solution satisfying

$$x_B = f + \sum \left( r^j x_j : j \in N \right),$$

where $B$ is the set of basic variables satisfying $B \subseteq \{1, \ldots, n\}$, $|B| = m$, $N$ is the set of non-basic variables defined as $N = \{1, \ldots, n\} \setminus B$, and where

$f, r^j \in \mathbb{Q}^m, \forall j \in N,\ f \geq 0$. The basic solution is $x^* = (x_B, x_N) = (f, 0)$, and is an optimal solution to (1) if and only if $x_i^* \in \mathbb{Z}, \forall i \in I \cap B$. If not, then one might try to find a valid inequality cutting off $x^*$ from the feasible region of (2). For simplicity, we will assume that $B \cap I = B$. One possibility is to consider the following relaxation of the feasible region of (1):

$$R_f(\tilde{r}^i : i \in N) := \left\{(x', s) \in \mathbb{Z}^m \times \mathbb{R}_+^N : x' = f + \sum(\tilde{r}^i s_i : i \in N)\right\}, \qquad (3)$$

where $\tilde{r}^i = r^i$ for $i \in N \setminus I$, $\tilde{r}^i = r^i - a^i$ for $i \in N \cap I$ for some fixed $a^i \in \mathbb{Z}^m$. Note that if $f$ is fractional, then $(f, 0) \notin R_f(\tilde{r}^i : i \in N)$. This relaxation was considered in [2, 11] for the case $m = 2$.

Gomory and Johnson [12] suggested relaxing (3) to an infinite-dimensional space; following the notation in [6]; it can be described as:

$$R_f := \left\{(x, s) \in \mathbb{Z}^m \times \mathbb{R}_+^{\mathbb{Q}^m} : x = f + \sum_{r \in \mathbb{Q}^m} r s_r,\ s \text{ with finite support}\right\}, \qquad (4)$$

This is called the *infinite relaxation* and its only parameter is $f \in \mathbb{Q}^m$. Note that any valid inequality for (4) yields a valid inequality for (3), which in turn can be transformed into a valid inequality for (1).

Borozan and Cornuéjols [6] studied minimal valid inequalities for (4), proving the following:

**Theorem 2.1 ([6]).** *If $f \notin \mathbb{Z}^q$, then any minimal valid inequality that cuts off $(f, 0)$:*

*i. Is of the form $\sum_{finite} \psi(r) s_r \geq 1$.*
*ii. $\psi$ is positive, sub-additive, homogeneous, convex and piecewise linear.*
*iii. If $B_\psi = \{x \in \mathbb{R}^p : \psi(x - f) \leq 1\}$, then $B_\psi$ is convex, with no integral point in its interior. Furthermore $f \in B_\psi$.*
*iv. If $\psi$ is finite, then $\psi$ is a continuous non-negative homogeneous convex piecewise linear function with at most $2^q$ pieces.*
*v. If $\psi$ is finite, then $f \in {B_\psi}^\circ$ and $B_\psi$ is a polyhedron of at most $2^q$ facets, and each of its facets contains an integral point in its relative interior.*

## 3. Selecting a subclass of valid inequalities

### *3.1. The Sets:*

Since we are interested in generating minimal inequalities, in the light of Theorem 2.1, we restrict ourselves to families of bounded, maximal, lattice-free polyhedra $B_\psi \subset \mathbb{R}^n$ satisfying Theorem 2.1.v.

Although the characterization of all maximal lattice-free convex sets in the plane is known [16], such a characterization is unknown for arbitrary dimensions. For this reason we test three simple families.

Finally, note that for a valid inequality $\psi$, the set $B_\psi$ is the set of fractional points around $f$ forbidden by the inequality. This suggest that the volume of $B_\psi$ may provide a quality measure of the strength of the implied cut.

**Definition 3.1** ($T1_n$)**.** *We define*

$$T1_n := \left\{x \in \mathbb{R}^n : x \geq 0,\, e^t x \leq n\right\},$$

*where $e$ is the vector of all ones.*

It is easy to see that $T1_n$ is defined by $n+1$ inequalities, each of them containing one integer point in its relative interior; ${T1_n}^\circ \cap \mathbb{Z}^n = \emptyset$; and its volume is $n^n/n!$. i.e. it is a bounded, convex, maximal, lattice-free set in $\mathbb{R}^n$.

**Definition 3.2** ($G_n$)**.** *We define*

$$G_n := 1/2e + \left\{x : \delta^t x \leq n/2,\, \forall \delta \in \{-1,1\}^n\right\}$$

Note that $G_n$ is just the cross polytope in $\mathbb{R}^n$ containing the 0-1 hypercube, it is defined by $2^n$ inequalities, each of them containing exactly one integer point in its relative interior; and its volume is $n^n/n!$. i.e. it is a bounded, convex, maximal, lattice-free set in $\mathbb{R}^n$.

**Definition 3.3** ($T2'_n$)**.** *We define*

$$T2'_n := \left\{ x : \begin{array}{lrl} (R_j) & \sum_{i=1}^{j-1} x_i \leq & x_j,\, j = 1..n \\ (R_{n+1}) & e^t x \leq & 2^n - 1 \end{array} \right\},$$

*and the vectors $\{v_{k,n}\}_{k=1}^{n+1}$ where $(v_{k,n})_i = 0$ if $i < k$, $(v_{k,n})_k = 2^k(1-2^{-n})$ and $(v_{k,n})_i = 2^{i-1}(1-2^{-n})$ if $i > k$.*

It is easy to check that $v_{k,n}$ is the vertex of $T2'_n$ defined by the intersection of all constraints but $(R_k)$. Moreover, its volume is $\left(2^{\frac{n+1}{2}} - 2^{\frac{1-n}{2}}\right)^n/n!$, which is considerable bigger than the volume of $T1_n$ as $n$ grows.

**Proposition 3.1.** *There is no point $z \in \mathbb{Z}^n$ in the interior of $T2'_n$ and each facet has an integer point in its relative interior.*

*Proof.* Define $T2_n = T2'_n + v_o$, where $(v_o)_i = 1 - 2^{i-1}$. Since $v_o$ is an integer vector, is enough to prove that $T2_n$ is lattice free and that each facet has an integer point in its relative interior.

Note that $T2_n$ is the set of points $x \in \mathbb{R}^n$ satisfying $\sum_{i=1}^{j-1} x_i - x_j \leq j-1$, for $j = 1, \ldots, n$ and $e^t x \leq n$.

We proceed by induction on $n$. By definition $T2_1 = \text{Conv\_Hull}\{0,1\}$, thus the statement holds for $n = 1$. Now, suppose it holds for $n-1$. First, note that $(v_o + v_{k,n})_1 \in \{0, 2(1-2^{-n})\}$, then $\text{proj}_{x_1}(T2_n) = [0, 2-2^{1-n}]$. Define $S_1 := (\{x : x_1 = 0\} \cap T2_n \cap \mathbb{Z}^n)$ and $S_2 := (\{x : x_1 = 1\} \cap T2_n \cap \mathbb{Z}^n)$. Then $T2_n \cap \mathbb{Z}^n$ can be re-written as $S_1 \cup S_2$, but all points in $S_1$ satisfy $R1$ at equality. Moreover, all points in $\{(0,y) : y \in T2_{n-1} \cap \mathbb{Z}^{n-1}\}$ only satisfy $R1$ at equality. For points in $S_2$, note that $\{x : x_1 = 1\} \cap T2_n = \{x \in \mathbb{R}^n : x = (1,y), y \in T2_{n-1}\}$, then, by our hypothesis, all integer points in $S_2$ satisfy some constraint of $T2_n$. ∎

*3.2. Properties of the selected families:*

For the case $n = 2$, Cornuéjols and Margot [7] proved that, assuming $f \in B_\psi{}^\circ$, $T1_2, T2_2$ (called triangle inequalities in [2]) are facet defining for $R_f$, but that $G_2$ is not. However, it is easy to prove that a small perturbation of any of the inequalities defining $G_2$, makes the resulting inequality a facet of $R_f$. This observation, and the limited numerical precision of floating point representation, justify, from a practical point of view, overlooking the fact that $G_2$ does not define a facet of $R_f$ for $n = 2$.

For $n > 2$, it is easy to extend the results of [7], and prove that all sets in both $T1_n$ and $T2_n$ are facets of $R_f$. However, whether a similar extension can be made for $G_n$, is unclear.

Note also that the sets $T1_n$, $T2_n$ and $G_n$ contain the 0-1 hypercube in $\mathbb{R}^n$, thus, if we rotate in 180 degrees any axis around $\frac{1}{2}e$, or if we permute variables, we also obtain maximal lattice-free sets. $G_n$, is not affected by any of this transformations; $T1_n$ generate $2^n$ different sets; and $T2_n$ generate $2^n(n-1)!$ different sets. We will abuse notation, and call, each individual set a *configuration*, and the family of sets will be denoted, for fixed $n$, $G_n$, $T1_n$ and $T2_n$ respectively.

## 4. Separating the inequalities

*4.1. From lattice-free sets to cuts*

Given a convex lattice-free set $S$, $f \in S^\circ$, by Theorem 2.1, if $\lambda$ is such that $\lambda r \in \partial S$, then $\psi_S(r) = 1/\lambda$. In our case, each set $S$ is given by a set of inequalities $\{a_i x \leq b_i\}_{i=1}^m$, thus $\lambda = \min\{\lambda_i : \lambda_i \geq 0,\, i = 1, \ldots, m\}$ where $\lambda_i := (b_i - a_i f)/a_i r$, and where we assume that $\pm 1/0 = \pm\infty$. In other words, $\psi_S(r) = \max\{a_i r/(b_i - a_i f) : a_i r \geq 0,\, i = 1, \ldots, m\}$.

Given a cut, $ax \geq 1$, we use as a quality measure $1/\|a\|_1$. Thus, given $f$ and a family of sets $\mathcal{S}$, the problem of finding the best inequality, reduce to computing $a$ for all configurations and keep the best cut. Although the work is exponential in $n$, for $n$ small enough this is not a problem. The process of iterating over $\{-1, 1\}^n$ was implemented using gray-code enumeration [14], and the process of iterating over all permutations of $n$ was implemented using the plain changes algorithm [14]. The main advantage of these algorithms is that the difference between one configuration and the next is small (no more than two elements differ). Thus allowing to save part of previous computations.

*4.2. Getting relevant relaxations from a tableau*

An important problem is to choose an appropriate relaxation of the form (3). In principle any set of tableau rows with integer basic variables, at least one of them fractional, would provide a relaxation from where we can derive a separating cut. In our implementation, given a target number of tableau rows $k$, we sort all integer variables by fractional value, and compute the associated tableau row for the variable. This tableau is computed as a linear combination of the original constraints. Tableau rows considered numerically unstable, are discarded from the list. We iterate until no more fractional variables are available, or the target number of tableau rows $k$ has been selected.

Once a set of tableau-rows has been selected, we need to define the coefficients $a^i$ for the non-basic integer variables. In principle, the only restriction is that $a^i \in \mathbb{Z}^k$, however, since we prefer smaller coefficients in the cut, we would like the resulting vector $r^i - a^i$ to be small. For simplicity, in our case we choose them as the integer part of the vector $r^i$. Note that the problem of choosing the best $a^i$ is related to the problem studied in [8]. This define a relaxation as in (3).

For each family of inequalities, we keep up to five cuts according to the previously defined quality measure, and add them to the current relaxation using a cut callback. The procedure may be called several times during the optimization process, but cuts will be added only at the root node.

*4.3. Some numerical considerations*

In order to improve the numerical stability of the resulting cuts, we discard tableau rows $\overline{a}x = \overline{b}$ such that $\frac{\max\{|\overline{a}_i|\}}{\min\{|\overline{a}_i|\}} > 4000$. All coefficients with absolute value below $10^{-6}$ are considered as zero. We test that each computed tableau row has one basic variable, and that the basic coefficient is one, i.e. $|a_{basic} - 1| \leq 1e - 6$. We also discard cuts $ax \geq 1$ such that $\frac{\max\{|a_i|\}}{\min\{|a_i|\}} > 32000$.

## 5. Computational Results

Our main aim is to show that cuts derived from the infinite group relaxation can have a positive impact in practice on a wide set of test instances. With this objective in mind, we use as base reference CPLEX 11.0, and compare both the LP bound obtained at the root node after cut generation, and the final performance of the branch and bound algorithm.

*5.1. The Instances:*

Our set of MIP instances contains all MIPLIB 3.0 [3], MIPLIB 2003 [1], and other problems from the literature, the resulting test set contains 173 problems. Since we are interested in the effect of cutting planes, we discarded all problems (51) whose root LP gap (measured as the gap between the optimal and the best LP value at the root node) was below 0.5% on all configurations. Furthermore, we discarded all problems (11) where the root LP time (with cutting enabled) for all configurations was over 30 minutes. This leaved us with 111 test cases.

*5.2. Root LP Comparisons:*

We denote by $z^b_{LP}$ the LP value obtained with CPLEX's default configuration with aggressive pre-processing enabled; $z_{IP}$ the optimal or best know solution for a problem; and $z^*_{LP}$ the best root LP bound found among all configurations. We define two quality measures: closed gap at root node (CGAP) and relative loss against best achieved bound (RLOSS) as

$$CGAP : \frac{z_{LP} - z^b_{LP}}{z_{IP} - z^b_{LP}}, \qquad RLOSS : \frac{z_{LP} - z^*_{LP}}{\max\{1, |z_{IP}|, z_{IP} - z^b_{LP}\}}.$$

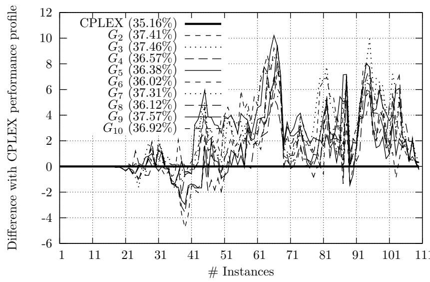


Figure 1: Difference of CGAP for $G_n$ configurations against CPLEX CGAP

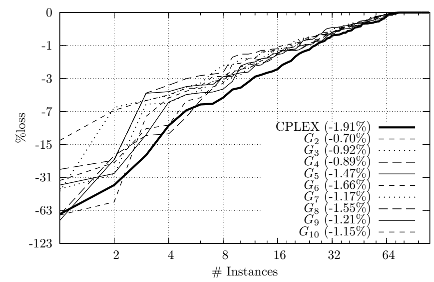


Figure 2: Performance profile of RLOSS for $G_n$ configurations

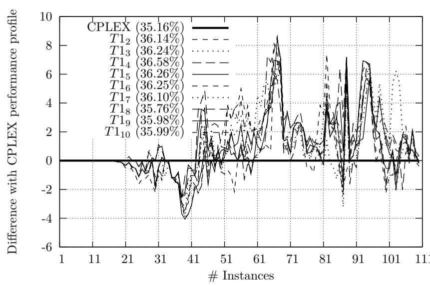


Figure 3: Difference of CGAP for $T1_n$ configurations against CPLEX CGAP

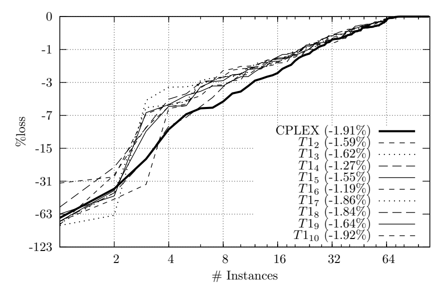


Figure 4: Performance profile of RLOSS for $T1_n$ configurations

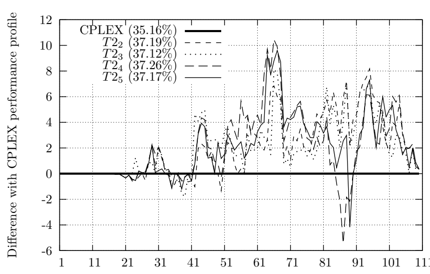


Figure 5: Difference of CGAP for $T2_n$ configurations against CPLEX CGAP

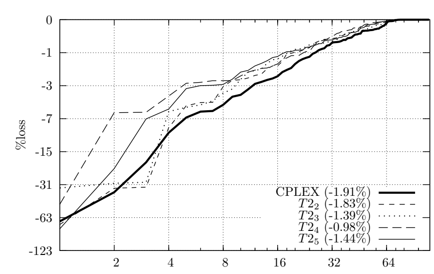


Figure 6: Performance profile of RLOSS for $T2_n$ configurations

We tested $G_n$ and $T1_n$ for $n = 2, \ldots, 10$ and $T2_n$ for $n = 2, \ldots, 5$. In each round of cuts we added up to five cuts, and we added up to 100 cuts in total, all of them at the root node. As base comparison, we tested CPLEX defaults with aggressive pre-solve enabled and with a cut-factor limit of ten, i.e. we limit the total number of cuts to be added (from CPLEX and from our procedure) to be up to nine times the number of original constraints. These computations where carried out in a Linux 2.6.9 machine with 16Gb of RAM, and two Quad-Core Intel Xeon E5420 processor.

Figures 1, 3 and 5 show the difference of performance profiles for CGAP using as base CPLEX's results, they also show the arithmetic average of CGAP. Figures 2, 4 and 6 show the performance profile for RLOSS using a logarithmic scale on both plot axes, they also show the arithmetic average of RLOSS.

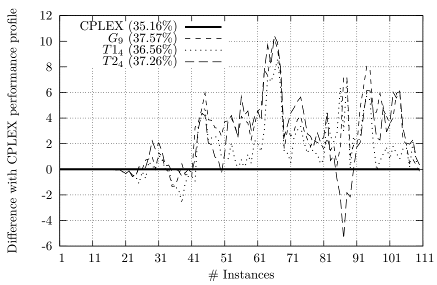


Figure 7: Difference of CGAP for best configurations against CPLEX CGAP

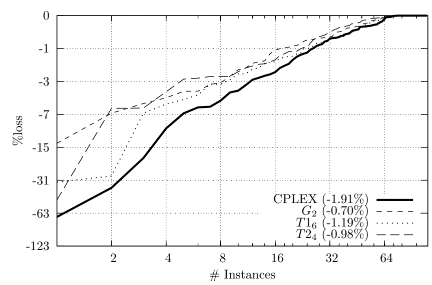


Figure 8: Performance profile of RLOSS for best configurations

Figure 7 and 8 summarizes the performance of the best configuration for each family of lattice-free sets. If we look at CGAP, the best configuration is $G_9$, with an average of 37.57% of closed gap at root node, which is 2.51 points better than CPLEX. For RLOSS, CPLEX is dominated by $G_2, T1_6$ and $T2_4$. The best configuration is $G_2$, with an average of -0.70%, which is 1.21 points better than CPLEX.

While in most cases the results of $G_2, T1_2$ and $T2_2$ where dominated by some $G_n, T1_n$ and $T2_n$, they were not always dominated, thus proving that looking into relaxations with more rows can be valuable, but at the same time, showing that this is not always the case. Identifying when either case holds remains unanswered.

Another interesting point to note is that, in general, $G_n$ was better than $T2_n$, which in turn was better than $T1_n$, even though $G_n$ is never a facet of $R_f$.

### 5.3. Branch and Cut Comparisons

Our main goal here is to show that the cuts may be helpful when evaluated within a branch and cut scheme, and at the same time see the effect of looking at relaxations with more than two rows.

For those instances that can be solved by at least one configuration, the quality measure will be total running time. For those instance that could not be solved by any configuration, the quality measure will be the closed gap to the optimal (or best known) solution.

To simplify comparisons and avoid interference from other components within B&B, we used best bound node selection rule for all runs, and disable CPLEX internal heuristics for feasible solutions.

We tested configuration $G_1, \ldots, G_{10}$, $T1_1, \ldots, T1_{10}$ and $T2_1 \ldots, T2_{10}$ and CPLEX defaults. Each instance was run under each configuration with a running time limit of five hours. These computations where carried out in a Linux 2.6.9 machine with 16Gb of RAM, and two Quad-Core Intel Xeon CPU E5420.

Out of the total 111 instances, five instances failed (in more than twenty tests) due to memory limits; we discarded them to have fairer comparisons. Another group of eleven instances were solved in under five seconds by all tests (but $T2_7, \ldots, T2_{10}$), and where discarded from the set of instances (to avoid comparisons of very small running times). The rest where separated into two groups: those that were solvable within five hours by at least one test (52), and those that could not be solved by any configuration (43).

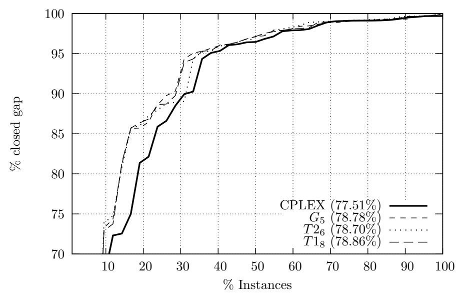


Figure 9: Time performance on finished instances and geometric average

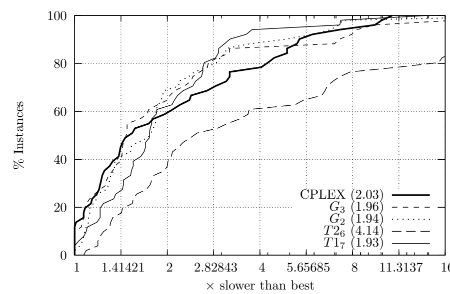


Figure 10: Closed gap performance on unfinished instances

Figure 10 shows the best time performance on solved instances; while figure 9 show the best closed gap performance on unsolved instances. Table 1 show the results for all tested configurations.

On instances where at least one test finished, we can see that both $G_2$ and $T1_7$ are faster (in geometric mean) by 5% than CPLEX; and if we look to plain average running time, $G_3, T1_7$ and $T2_6$ are 20%, 26% and 24% faster, respectively, than CPLEX. In the rest of the cases, after five hours of running time, we could get between 1.4% and 1.6% better bounds for most tested configurations.

## 6. Final Remarks and Conclusions

We have shown that even simple subclasses of inequalities derived from the infinite relaxation can have a positive impact both on overall branch and cut performance, and on the closed gap at the root node.

These results also point towards both trying to identify important classes of inequalities for $R_f$ for higher dimensions, and to find good computational implementation choices for them. Moreover, the question of when a given relaxation is better suited to generate cuts is still open.

| Geo. avg. closed gap | | | | | Avg. closed gap | | | | |
|---|---|---|---|---|---|---|---|---|---|
| t | CPLEX | G | T1 | T2 | t | CPLEX | G | T1 | T2 |
| 2 | 77.51 | 78.27 | 78.83 | 76.43 | 2 | 87.03 | 87.85 | **88.42** | 86.51 |
| 3 | 77.51 | 77.88 | 78.76 | 78.13 | 3 | 87.03 | 87.42 | 88.38 | 87.62 |
| 4 | 77.51 | 78.05 | 78.25 | 78.53 | 4 | 87.03 | 87.51 | 87.78 | 88.12 |
| 5 | 77.51 | **78.78** | 78.05 | 78.06 | 5 | 87.03 | **88.39** | 87.57 | 87.60 |
| 6 | 77.51 | 78.25 | 77.06 | **78.70** | 6 | 87.03 | 87.88 | 86.56 | **88.27** |
| 7 | 77.51 | 78.27 | 78.60 | 78.14 | 7 | 87.03 | 87.81 | 88.21 | 87.70 |
| 8 | 77.51 | 78.50 | **78.86** | 78.25 | 8 | 87.03 | 88.09 | 88.41 | 87.79 |
| 9 | 77.51 | 78.41 | 78.41 | 67.47 | 9 | 87.03 | 87.97 | 88.00 | 82.38 |
| 10 | 77.51 | 70.55 | 78.45 | 55.83 | 10 | 87.03 | 85.95 | 88.03 | 73.66 |

| Geo. avg. slower than best | | | | | Avg. running time | | | | |
|---|---|---|---|---|---|---|---|---|---|
| t | CPLEX | G | T1 | T2 | t | CPLEX | G | T1 | T2 |
| 2 | 2.03 | **1.94** | 2.20 | **2.21** | 2 | 2616 | 2114 | 2258 | 2775 |
| 3 | 2.03 | 1.96 | 2.18 | 2.25 | 3 | 2616 | **2087** | 2054 | 2313 |
| 4 | 2.03 | 2.17 | 2.13 | 2.48 | 4 | 2616 | 2433 | 2109 | 2263 |
| 5 | 2.03 | 2.13 | 1.98 | 2.64 | 5 | 2616 | 2413 | 1957 | 2258 |
| 6 | 2.03 | 2.02 | 2.22 | 4.14 | 6 | 2616 | 2302 | 2424 | **1971** |
| 7 | 2.03 | 2.21 | **1.93** | 12.43 | 7 | 2616 | 2447 | **1915** | 2381 |
| 8 | 2.03 | 2.28 | 2.23 | 66.87 | 8 | 2616 | 2522 | 2452 | 7975 |
| 9 | 2.03 | 2.01 | 2.11 | 205.16 | 9 | 2616 | 2456 | 1922 | 18000 |
| 10 | 2.03 | 2.17 | 2.24 | 231.26 | 10 | 2616 | 2505 | 2309 | 18000 |

Table 1: Summary of results for all tests: on columns, different configurations; on rows, maximum tableau rows used in the relaxation.

There are many possibilities to explore, like adding several cuts in every iteration from different relaxations, choosing several sets of tableau rows to work on, choosing different ground sets $B_\psi$, and separate maximally violated inequalities for the current relaxation instead of using fixed templates from the infinite relaxation.

Questions like how good is the infinite relaxation closure, or even an approximation of it, are still open.

## 7. Acknowledgments

Special thanks to an anonymous referee for his or her criticism and suggestions, which led to a much improved presentation of this paper. We also thanks the generous support of FONDECYT grant 1070749 and ICM grant P05-004F.

## References

[1] T. Achterberg, T. Koch, and A. Martin. MIPLIB 2003. *Operations Research Letters*, 34(4):1–12, 2006.

[2] K. Andersen, Q. Louveaux, R. Weismantel, and L. A. Wolsey. Inequalities from two rows of a simplex tableau. In M. Fischetti and D. P. Williamson,

editors, *IPCO*, volume 4513 of *Lecture Notes in Computer Science*, pages 1–15. Springer-Verlag, 2007.

[3] R. E. Bixby, E. A. Boyd, and R. R. Indovina. MIPLIB: A test set of mixed integer programming problems. *SIAM News*, 25:16, 1992.

[4] R. E. Bixby, M. Fenelon, Z. Gu, E. Rothberg, and R. Wunderling. MIP: Theory and practice - closing the gap. In *Proceedings of the 19th IFIP TC7 Conference on System Modelling and Optimization*, pages 19–50, Deventer, The Netherlands, 2000. Kluwer, B.V.

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

[6] G. Cornuéjols and V. Borozan. Minimal valid inequalities for integer constraints. George Nemhauser Symposium, Atlanta, July 2007.

[7] G. Cornuéjols and F. Margot. On the facets of mixed integer programs with two integer variables and two constraints. *Mathematical Programming*, 120(2):429–456, 2009.

[8] S. S. Dey and L. A. Wolsey. Lifting integer variables in minimal inequalities corresponding to lattice-free triangles. In A. Lodi, A. Panconesi, and G. Rinaldi, editors, *IPCO 2008*, volume 5035 of *Lecture Notes in Computer Science*, pages 463–475, 2008.

[9] D. Espinoza. Computing with multi-row gomory cuts. In A. Lodi, A. Panconesi, and G. Rinaldi, editors, *IPCO 2008*, volume 5035 of *Lecture Notes in Computer Science*, pages 214–224, 2008.

[10] R. E. Gomory. An algorithm for the mixed integer problem. Technical Report RM-2597, RAND Corporation, 1960.

[11] R. E. Gomory. Corner polyhedra and two-equation cutting planes. George Nemhauser Symposium, Atlanta, July 2007.

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

[13] ILOG CPLEX Division, Incline Village, Nevada, 89451, USA. *CPLEX 11.0 Reference Manual*, 2007.

[14] D. E. Knuth. *The Art of Computer Programming*, volume 4. Addisoin-Wesley, 1st. edition, February 2005.

[15] Y. Li and J.-P. P. Richard. Cook, kannan and schrijver's example revisited. *Discrete Optimization*, 5(4):724–734, 2008.

[16] L. Lovász. Geometry of numbers and integer programming. In M. Iri and K. Tanabe, editors, *Mathematical Programming: Recent Developments and Applications*, pages 177–210. Springer, 1989.