K-Cuts: A Variation of Gomory Mixed Integer Cuts from the LP Tableau

Gérard Cornuéjols Yanjun Li Dieter Vandenbussche

Graduate School of Industrial Administration
Carnegie Mellon University, Pittsburgh, PA 15213, USA

Graduate School of Industrial Administration
Carnegie Mellon University, Pittsburgh, PA 15213, USA

Department of Industrial and Systems Engineering
Georgia Institute of Technology, Atlanta, GA 30332, USA

gc0v@andrew.cmu.edu yanjun@andrew.cmu.edu dieter@akula.isye.gatech.edu

---

For an integer program, a *k-cut* is a cutting plane generated by the Gomory mixed integer procedure from a row of the LP tableau after multiplying it by a positive integer k. With this terminology, Gomory mixed integer cuts are just 1-cuts. In this paper, we compare the k-cuts ($k \geq 2$) with Gomory mixed integer cuts. In particular, we prove in the pure case that with exactly 50% probability the k-cuts perform better variable-wise than the Gomory mixed integer cuts, and vice versa. Some computational experiments on knapsack problems are reported to illustrate this property.
(*Integer Programming; LP Tableau; Gomory Mixed Integer Cut; K-cut*)

---

1. Introduction

In the late fifties and early sixties, Gomory introduced two kinds of general cutting planes, fractional cuts and mixed integer cuts [9, 10, 11], for solving

integer programming problems. Later on, extensive work was done on the properties of Gomory cuts and the convergence of Gomory cutting plane methods [3, 5, 7, 8, 14, 22, 23]. The potential of Gomory cuts in computation was also explored [20, 21]. The advantage of Gomory cuts within a branch-and-cut framework was computationally shown in [2].

Glover [8] proposed to linearly combine the original equations from the linear programing tableau with Gomory cuts. The combination allows two free parameters to be fixed. Glover discussed how to specify the parameter values so that the resulting cuts "not only limit the feasible set more restrictively than the method of integer form, but satisfy other criteria as well".

In a series of papers [12, 15, 16, 18] in the late sixties and early seventies, Gomory and Johnson showed how to generate cutting planes from subadditive functions and the group problem. Recent talks have been given on this subject [13]. Cutting plane generation from subadditive functions is a general framework. The k-cuts in this paper can be viewed as a particular case of this framework; the subadditive functions for k-cuts are the simplest ones among the possible subadditive functions. Nevertheless, it is interesting to study this particular case, because it comes from a familiar and very natural operation.

In a section of their book [7] Garfinkel and Nemhauser discuss the possibility of obtaining stronger cuts than the Gomory fractional cuts by first multiplying a tableau row by a nonzero factor before generating the fractional cut. The potential multipliers could be integral or non-integral numbers. For integral multipliers, the larger the fractional part of the right-hand-side of the resulting tableau row, the stronger the resulting cut becomes in most of the cases. For non-integral multipliers, the situation turns out to be complicated.

Dawande and Pulleyblank [6] considered the cone formed at the LP optimum by the normals of the tight constraints at the optimum. The direction vector (the normal to the cut hyperplane) of the Gomory fractional cut lies in this cone. The direction vector is integral. In general, there are several integral vectors in the cone. After multiplying the tableau rows by integral multipliers, precisely which integral vectors in the cone are the direction vectors of the cuts? Dawande and Pulleyblank have done some work on characterizing these directions in order to better understand the Gomory fractional cuts from a geometrical point of view.

Günlük and Pochet [17] described a simple cutting-plane algorithm that uses a "mixing procedure" to generate valid inequalities. First, they generate 10 base inequalities from each tableau row by multiplying the row with the multipliers $\delta = 1, 2, 3, 4, 5$ and $\delta = -1, -2, -3, -4, -5$. And then they apply

their "strengthening procedure" to the collection of the base inequalities to obtain cuts and improve the relaxation bound for general integer programs.

In this paper we focus on a variation of Gomory mixed integer cuts for pure integer programs. Rather than generating Gomory mixed integer cuts from the LP tableau rows directly, we first multiply the tableau rows by an integer k and then generate the cuts from the new rows according to the Gomory mixed integer procedure. We call them *k-cuts* in this paper. A comparison of k-cuts ($k \geq 2$) with the usual Gomory mixed integer cuts $k = 1$) is given in Section 3. It is proved in the pure case that with exactly 50% probability the k-cuts perform variable-wise better than the Gomory mixed integer cuts. These k-cuts provide a variety of cuts different from Gomory mixed integer cuts and can cut off the relaxation set from different angles. Computational results on knapsack problems are given in Section 4 to illustrate that the k-cuts have the same power as Gomory mixed integer cuts in terms of improving the relaxation bound. If we impose all the k-cuts ($1 \leq k \leq M$) as constraints, we see empirically that they are effective as a whole in restricting the feasible set of the LP relaxation of the knapsack problem when M becomes larger. Surprisingly, the same does not appear to be true for integer programs with many constraints, as illustrated at the end of Section 4. An extension of k-cuts to mixed integer programs is discussed in the last section.

2. Gomory mixed integer cuts and k-cuts

In this section, we start with the definition and properties of k-cuts, and we give two examples. Finally, we introduce a graphic tool that is helpful when we compare the cuts in the next section.

Consider the pure integer program (IP)

$$\begin{aligned} & \text{Min } cx \\ \text{s.t. } & Ax = b, \\ & x \in Z^n_+, \end{aligned}$$

where A is a rational matrix, and c and b are rational vectors.

After solving the LP relaxation of (IP), we have the following basic solution

$$x_i = \bar{a}_{i0} + \textstyle\sum_{j \in J} \bar{a}_{ij}(-x_{ij}) \quad \text{for all } i \in I, \tag{1}$$

where I and J denote the sets of basic and non-basic variables.

Write $\bar{a}_{ij} = \lfloor \bar{a}_{ij} \rfloor + f_{ij}$ ($i \in I$ and $j \in J$) and $\bar{a}_{i0} = \lfloor \bar{a}_{i0} \rfloor + f_{i0}$ ($i \in I$). Assume $f_{i0} > 0$ for some i. The *Gomory fractional cut* (GFC) from that LP tableau row is

$$\sum_{j\in J} f_{ij}x_j \geq f_{i0}. \tag{2}$$

The *Gomory mixed integer cut* (GMIC) from the LP tableau row is:

$$\sum_{(j:f_{ij}\leq f_{i0})} f_{ij}x_j + \frac{f_{i0}}{1-f_{i0}} \sum_{(j:f_{ij}>f_{i0})}(1-f_{ij})x_j \geq f_{i0}. \tag{3}$$

Rather than directly generating the GMIC from the LP tableau row, we could first multiply both sides of the row by some nonzero integer $k \neq 1$ and then apply the Gomory mixed integer procedure as above to obtain the corresponding inequality:

$$\sum_{(j:\tilde{f}_{ij}\leq \tilde{f}_{i0})} \tilde{f}_{ij}x_j + \frac{\tilde{f}_{i0}}{1-\tilde{f}_{i0}} \sum_{(j:\tilde{f}_{ij}>\tilde{f}_{i0})}(1-\tilde{f}_{ij})x_j \geq \tilde{f}_{i0}, \tag{4}$$

where $\tilde{f}_{i0} = \tilde{a}_{i0} - \lfloor \tilde{a}_{i0} \rfloor > 0$, $\tilde{f}_{ij} = \tilde{a}_{ij} - \lfloor \tilde{a}_{ij} \rfloor > 0$ $(j \in J)$, $\tilde{a}_{i0} = k\bar{a}_{i0}$ and $\tilde{a}_{ij} = k\bar{a}_{ij}$ $(j \in J)$. We call the inequality in (4) a *k-cut* generated from the LP tableau row.

LEMMA 1. *The k-cut is a valid inequality for (IP).*

PROOF. Multiplying both sides of the LP tableau row by a nonzero integer $k \neq 0$, we have $kx_i = \tilde{a}_{i0} + \sum_{j\in J} \tilde{a}_{ij}(-x_{ij})$. Replacing kx_i by a new *integer* variable x_i', we get $x_i' = \tilde{a}_{i0} + \sum_{j\in J} \tilde{a}_{ij}(-x_{ij})$, which is an equation similar to (1). The standard derivation of GMIC applied to this equation shows that (4) is valid for IP. □

LEMMA 2. *The k-cut is equivalent to the* $(-k)$*-cut.*

PROOF. It is sufficient to show that the GMIC is equivalent to the (-1)-cut. The GMIC from the LP tableau row is shown in (3), which is

$$\sum_{(j:f_{ij}<f_{i0})} f_{ij}x_j + \sum_{(j:f_{ij}=f_{i0})} f_{i0}x_j + \frac{f_{i0}}{1-f_{i0}} \sum_{(j:f_{ij}>f_{i0})} (1-f_{ij})x_j \geq f_{i0}.$$

After multiplying -1 to the tableau row, the fractional parts of the coefficients in the new row are just $1-f_{ij}$ and $1-f_{i0}$. Therefore, the (-1)-cut is

$$\sum_{(j:f_{ij}>f_{i0})} (1-f_{ij})x_j + \sum_{(j:f_{ij}=f_{i0})} (1-f_{i0})x_j + \frac{1-f_{i0}}{f_{i0}} \sum_{(j:f_{ij}<f_0)} f_{ij}x_j \geq 1-f_{i0}.$$

We now see that there exists just a positive factor $\frac{f_{i0}}{1-f_{i0}}$ between the GMIC and the (-1)-cut. □

Now let us try to get some sense of the variety and efficiency of the k-cuts by analyzing two small examples.

EXAMPLE 1 (general integer case)

Consider the following integer programming problem:

$$\begin{aligned} \text{Max} \quad & 10x + 13y \\ \text{s.t.} \quad & 10x + 14y \leq 43, \\ & x,\ y \in Z_+. \end{aligned}$$

It is equivalent to the following IP after introducing an integer slack z:

$$\begin{aligned} \text{Min} \quad & y + z \\ \text{s.t.} \quad & x + 1.4y + 0.1z = 4.3, \\ & x,\ y,\ z \in Z_+. \end{aligned}$$

Obviously, the equality $x + 1.4y + 0.1z = 4.3$ is a row of the corresponding optimal LP tableau. The LP optimum is $(x, y, z) = (4.3, 0, 0)$.

All the possible k-cuts generated from this tableau row are summarized in the following table and Figure 1.

GMIC and 9-cut	$\frac{36}{7}y + 2z \geq 6$
2-cut and 8-cut	$3y + 2z \geq 6$
3-cut and 7-cut	$2y + 3z \geq 9$
4-cut and 6-cut	$2y + 3z \geq 4$
5-cut	$z \geq 1$

A picture of these k-cuts in the (y, z)-plane clearly shows that all the k-cuts $(2 \leq k \leq 8)$ either dominate the GMIC or are incomparable with the GMIC in this example. These k-cuts effectively shrink the feasible region of the LP relaxation by cutting off different parts. Worth mentioning is that the 3-cut is a facet to the convex hull of the feasible integer points in the (y, z)-plane. Together with the valid inequalities $14y + z \leq 43$ and $y \geq 0$, it gives a complete facet description of the convex hull of the feasible (y, z) integer points.

If we consider the following IP instead of the above one

$$\begin{aligned} \text{Min} \quad & y + z \\ \text{s.t.} \quad & x + 0.4y + 0.1z = 5.3, \\ & x,\ y,\ z \in Z_+, \end{aligned}$$

then all the k-cuts generated from the new optimum LP tableau row are the same as above. However, in this case the 3-cut and the 5-cut, together with $4y + z \leq 53$ and $y \geq 0$, give a complete facet description of the corresponding convex hull of the feasible (y, z) integer points. □

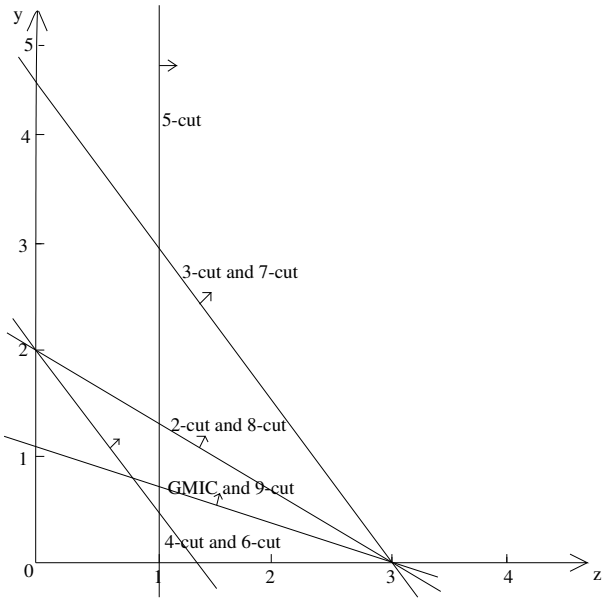

Figure 1: k-cuts in (y, z)-plane for Example 1

EXAMPLE 2 (0-1 case)

Consider the following 0-1 programming problem:

$$\begin{aligned} \text{Max} \quad & x_1 + x_2 \\ \text{s.t.} \quad & 10x_1 + 2x_2 \leq 11, \\ & 2x_1 + 10x_2 \leq 11, \\ & x_1 \leq 1, \ x_2 \leq 1, \\ & x_1, \ x_2 \in Z_+. \end{aligned}$$

After introducing two integer slacks w and z, we get the equivalent IP:

$$\begin{aligned} \text{Max} \quad & x_1 + x_2 \\ \text{s.t.} \quad & 10x_1 + 2x_2 + w = 11, \\ & 2x_1 + 10x_2 + z = 11, \\ & x_1 \leq 1, \ x_2 \leq 1, \\ & x_1, \ x_2 \ w, \ z \in Z_+. \end{aligned}$$

The GMICs generated from the optimum LP tableau rows $x_1 + \frac{5}{48}w - \frac{1}{48}z = \frac{11}{12}$ and $x_2 - \frac{1}{48}w + \frac{5}{48}z = \frac{11}{12}$ are

$$\begin{cases} 72x_1 + 120x_2 \leq 132, \\ 120x_1 + 72x_2 \leq 132. \end{cases}$$

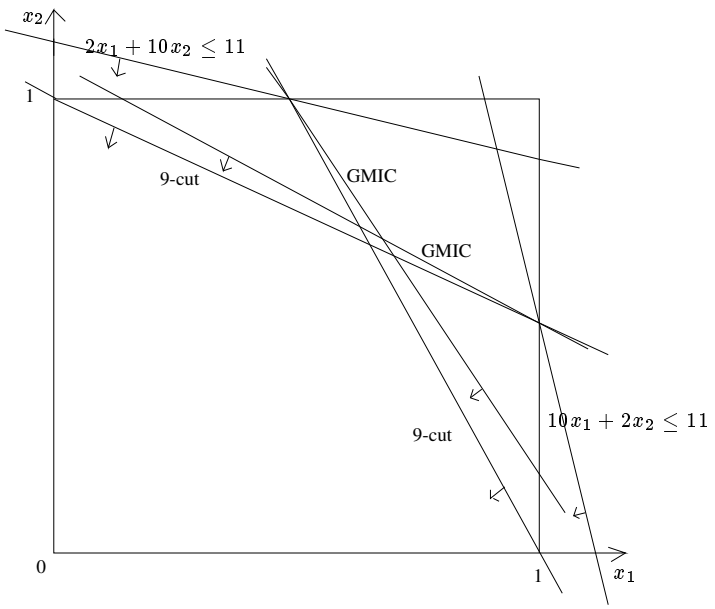

Figure 2: 9-cuts vs GMIC in (x_1, x_2)-plane for Example 2

Their respective boundary lines pass through the points "$(1, \frac{1}{2})$ and $(\frac{1}{6}, 1)$" and "$(1, \frac{1}{6})$ and $(\frac{1}{2}, 1)$" and they intersect at the point $(\frac{11}{16}, \frac{11}{16})$ in the (x_1, x_2)-space.

The 9-cuts generated from those two LP tableau rows are

$$\begin{cases} x_1 + 2x_2 \leq 2, \\ 2x_1 + x_2 \leq 2. \end{cases}$$

The corresponding boundary lines pass through the points "$(1, \frac{1}{2})$ and $(0, 1)$" and "$(1, 0)$ and $(\frac{1}{2}, 1)$" respectively and intersect at $(\frac{2}{3}, \frac{2}{3})$.

We see that these two 9-cuts respectively dominate the corresponding GMICs. In fact, together with the non-negativity constraints on x_1 and x_2, these two 9-cuts give a complete description of the elementary closure of the LP relaxation under the lift-and-project cuts. □

3. Comparing GMICs with k-cuts

In this section we are going to discuss the relation between Gomory mixed integer cuts and k-cuts in pure integer programs. The relation builds upon the following definition.

DEFINITION 1. *Let $\alpha x \geq \beta$ and $\bar{\alpha} x \geq \bar{\beta}$ be two cuts for (IP), where $\alpha, \bar{\alpha} \in R^n_+$, $\beta > 0$ and $\bar{\beta} > 0$. The cut $\alpha x \geq \beta$ is equally strong on x_j as the*

cut $\bar{\alpha}x \geq \bar{\beta}$*, if* $\alpha_j/\beta = \bar{\alpha}_j/\bar{\beta}$*. The cut* $\alpha x \geq \beta$ *is stronger on* x_j *than the cut* $\bar{\alpha}x \geq \bar{\beta}$*, if* $\alpha_j/\beta \leq \bar{\alpha}_j/\bar{\beta}$*. The cut* $\alpha x \geq \beta$ *is strictly stronger on* x_j *than the cut* $\bar{\alpha}x \geq \bar{\beta}$*, if* $\alpha_j/\beta < \bar{\alpha}_j/\bar{\beta}$*.* □

Obviously, if the cut $\alpha x \geq \beta$ is stronger on x_j than the cut $\bar{\alpha}x \geq \bar{\beta}$ for all $1 \leq j \leq n$, then the cut $\alpha x \geq \beta$ is stronger than the cut $\bar{\alpha}x \geq \bar{\beta}$, or in other words, $\alpha x \geq \beta$ dominates $\bar{\alpha}x \geq \bar{\beta}$.

Let $x_0 = \bar{a}_0 - \sum_{j \in J} \bar{a}_j x_j$ be an LP tableau row, where $\bar{a}_j = \lfloor \bar{a}_j \rfloor + f_j$, $\bar{a}_0 = \lfloor \bar{a}_0 \rfloor + f_0$ and $0 < f_0 < 1$. Let $\sum_{j \in J} \alpha_j x_j \geq \beta$ represent any of the three cuts generated from this tableau row: GFC as described in (2), GMIC as described in (3) and the k-cut as described in (4). Then $\alpha_j \geq 0$ $(j \in J)$ and $\beta > 0$. Given $0 < f_0 < 1$, α_j/β is in each case only a function of f_j. Actually, the function α_j/β for GFC is a linear function starting from origin, the function for GMIC can be plotted as a two-slope figure, and the function for a k-cut $(k \geq 2)$ is a piecewise linear function with a regular zig-zag feature. For instance, when $f_0 = 0.15$, the corresponding functions for GFC and GMIC are plotted in Figure 3. The function α_j/β corresponding to the 5-cut when $f_0 = 0.15$ is shown in Figure 4. This kind of picture allows to compare two inequalities by plotting the two functions and checking where one curve is below the other.

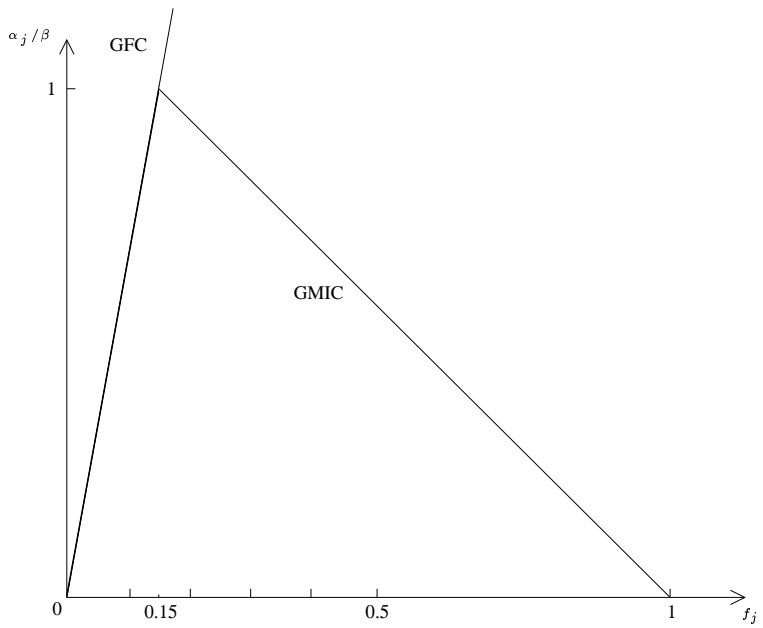

Figure 3: α_j/β as a function of f_j for GFC and GMIC

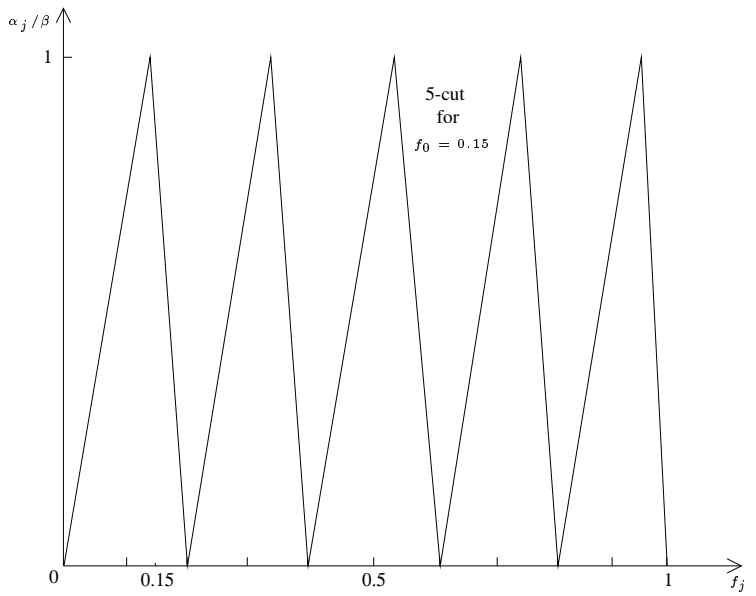

Figure 4: α_j/β as a function of f_j for a 5-cut

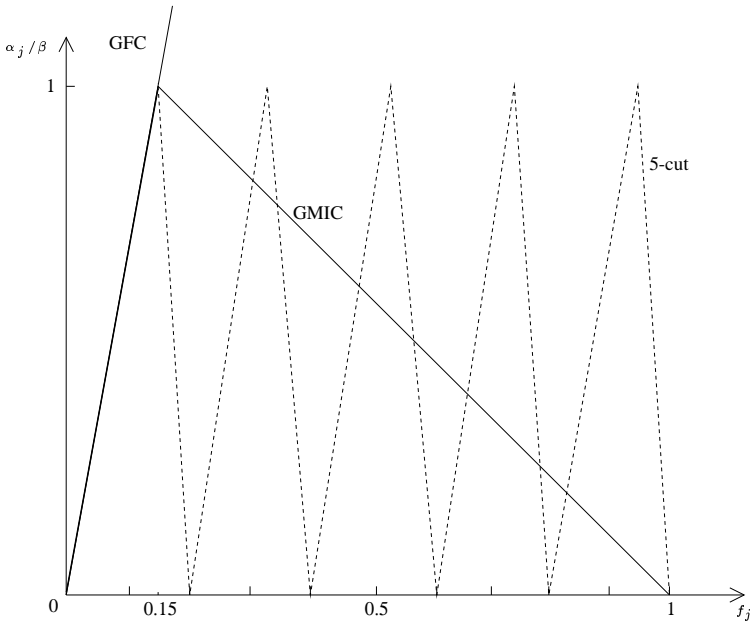

Figure 5: α_j/β as a function of f_j for GFC, GMIC and 5-cut

Figure 5 illustrates the relation among the functions α_j/β for GFC, GMIC and a 5-cut when $f_0 = 0.15$. Since $k = 5$ and $f_0 = 0.15$, we have $k < \lfloor \frac{1}{f_0} \rfloor$. We can observe from the figure: (1) The GFC, GMIC and the 5-cut are equally strong on x_j if $0 \leq f_j \leq f_0$; (2) The GMIC and the 5-cut are strictly stronger on x_j than GFC if $f_0 < f_j < 1$; (3) The GMIC is strictly stronger on x_j than the 5-cut in some of the sub-intervals of $(f_0, 1)$, while in the rest, except a few isolated points, 5-cut is strictly stronger on x_j than GMIC.

The observation from the example in Figure 5 can be extended to any $0 < f_0 < 0.5$ and $k < \lfloor \frac{1}{f_0} \rfloor$. The comparison is summarized in Table 1, where we use the numbers 1, 2 and 3 to rank the strength of GFC, GMIC and k-cut on x_j in a decreasing order.

It is easy to sum up from Table 1 the total length of the sub-intervals where the k-cut is strictly stronger on x_i than the GMIC. This length is equal to $\frac{1-f_0}{2}$. This exactly equals the total length of the sub-intervals where the GMIC is strictly stronger on x_i than the k-cut.

<table>
<tr><th colspan="3">f_j value</th><th>k-cut</th><th>GMIC</th><th>GFC</th></tr>
<tr><td colspan="3">$[0, f_0]$</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td colspan="3">$(f_0, \frac{1}{k})$</td><td>1</td><td>2</td><td rowspan="8">3</td></tr>
<tr><td rowspan="6">$[\frac{i}{k}, \frac{i+1}{k})$,
$1 \leq i \leq k-1$</td><td rowspan="3">$[\frac{i}{k}, \frac{i}{k} + f_0]$</td><td>$[\frac{i}{k}, \frac{i}{k} + \frac{k-i}{k} f_0)$</td><td>1</td><td>2</td></tr>
<tr><td>$\frac{i}{k} + \frac{k-i}{k} f_0$</td><td>1</td><td>1</td></tr>
<tr><td>$(\frac{i}{k} + \frac{k-i}{k} f_0, \frac{i}{k} + f_0]$</td><td>2</td><td>1</td></tr>
<tr><td rowspan="3">$(\frac{i}{k} + f_0, \frac{i+1}{k})$</td><td>$(\frac{i}{k} + f_0, \frac{i}{k-1} + \frac{k-i-1}{k-1} f_0)$</td><td>2</td><td>1</td></tr>
<tr><td>$\frac{i}{k-1} + \frac{k-i-1}{k-1} f_0$</td><td>1</td><td>1</td></tr>
<tr><td>$(\frac{i}{k-1} + \frac{k-i-1}{k-1} f_0, \frac{i+1}{k})$</td><td>1</td><td>2</td></tr>
</table>

Table 1: comparison among cuts when $0 < f_0 < 0.5$ and $1 \leq k \leq \lfloor \frac{1}{f_0} \rfloor$

THEOREM 1. *Let $x_0 = \bar{a}_0 - \sum_{j \in J} \bar{a}_j x_j$ be a LP tableau row, where $\bar{a}_j = \lfloor \bar{a}_j \rfloor + f_j$, $\bar{a}_0 = \lfloor \bar{a}_0 \rfloor + f_0$ and $0 < f_0 < 0.5$. Then we have*

(i) When $1 \leq k \leq \lfloor \frac{1}{f_0} \rfloor$, the k-cut generated from the row is stronger than the GFC generated from the row.

(ii) Assume that f_i is uniformly distributed in $[0, 1)$. When $2 \leq k \leq \lfloor \frac{1}{f_0} \rfloor$, the probability that the k-cut generated from the row is strictly stronger on x_i than the GMIC generated from the row is $\frac{1-f_0}{2}$, which is equal to the probability that the GMIC generated from the row is strictly stronger on x_i than the k-cut generated from the row. □

When we compare the k-cuts ($2 \le k \le \lfloor \frac{1}{f_0} \rfloor$) as a group with the GMIC, we have the following result.

THEOREM 2. *Assume that* $\frac{1}{K+1} \le f_0 < \frac{1}{K}$ *for some integer* $K \ge 2$ *and that* f_i *is uniformly distributed in* $[0, 1)$. *The probability that some* k*-cut generated from the row* ($2 \le k \le K$) *is strictly stronger on* x_i *than the GMIC generated from the row is* $\frac{(K-1)(1-f_0)}{K}$, *while the probability that the GMIC generated from the row is strictly stronger on* x_i *than all the* k*-cuts generated from the row* ($2 \le k \le K$) *is* $\frac{1-f_0}{K}$.

PROOF. From Table 1 we see that the GMIC and the k-cut ($2 \le k \le K$) are equally strong on x_j when $0 \le f_j \le f_0$.

Table 1 shows that the 2-cut is strictly stronger on x_j than the GMIC when $f_0 < f_j < \frac{1}{2}$. In the case of $k = 2$ and $i = 1$, the 2-cut is strictly stronger on x_j than the GMIC when $\frac{1}{2} \le f_j < \frac{1+f_0}{2}$.

Let $3 \le p \le K$. In the case of $k = p$ and $i = p - 2$, the p-cut is strictly stronger on x_j than the GMIC when $\frac{p-2+f_0}{p-1} < f_j < \frac{p-1}{p}$, while in the case of $k = p$ and $i = p - 1$ the p-cut is strictly stronger on x_j than the GMIC when $\frac{p-1}{p} \le f_j < \frac{p-1+f_0}{p}$. So the p-cut is strictly stronger on x_j than the GMIC when $\frac{p-2+f_0}{p-1} < f_j < \frac{p-1+f_0}{p}$.

If we put together all the intervals $(f_0, \frac{1}{2})$, $[\frac{1}{2}, \frac{1+f_0}{2})$ and $(\frac{p-2+f_0}{p-1}, \frac{p-1+f_0}{p})$ ($3 \le p \le K$), we see that, for any given value of $f_j \in (f_0, \frac{K-1+f_0}{K})$ except on $K - 2$ isolated points $\frac{p-1+f_0}{p}$ ($2 \le p \le K - 1$), there must exist some k-cut ($2 \le k \le K$) such that the k-cut is strictly stronger on x_j than the GMIC. The length of the interval $(f_0, \frac{K-1+f_0}{K})$ is $\frac{(K-1)(1-f_0)}{K}$.

It is not difficult to see in the case of $k = p$ and $i = p - 1$ ($2 \le p \le K$) that the GMIC is strictly stronger on x_j than the p-cut when $f_j \in (\frac{p-1+f_0}{p}, 1)$. Therefore, the GMIC is strictly stronger on x_j than all the k-cuts ($2 \le k \le K$) when $f_j \in (\frac{K-1+f_0}{K}, 1)$. The length of the interval $(\frac{K-1+f_0}{K}, 1)$ is $\frac{1-f_0}{K}$. □

Now let us compare the GMIC with a k-cut in the case $kf_0 > 1$.

In Figure 6 ($f_0 = 0.375$) and Figure 7 ($f_0 = 0.875$), we can respectively observe the relation between the functions α_j/β of the GMIC and the 4-cut. We have $\lceil kf_0 \rceil = \lceil 4 \times 0.375 \rceil = \lceil 1.5 \rceil = 2 < 4 = k$ in Figure 6 and $\lceil kf_0 \rceil = \lceil 4 \times 0.875 \rceil = \lceil 3.5 \rceil = 4 = k$ in Figure 7. In fact, these two figures effectively represent all the possible combinations of k and f_0 when $kf_0 > 1$.

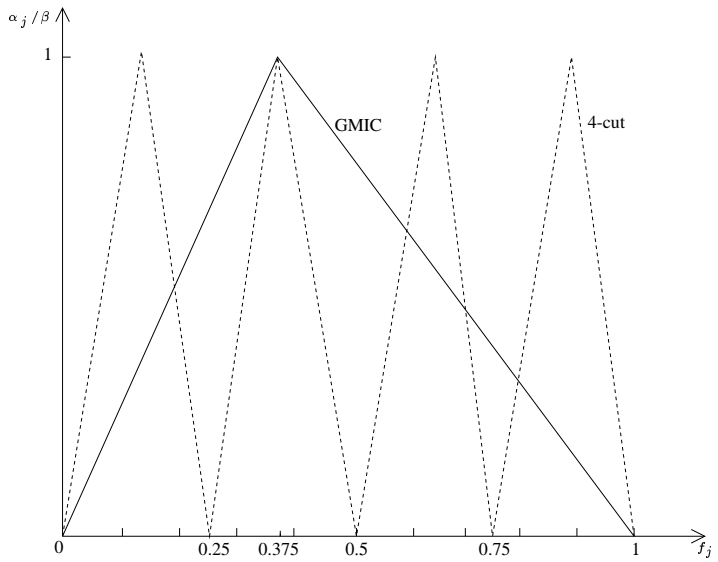

Figure 6: GMIC and 4-cut when $f_0 = 0.375$

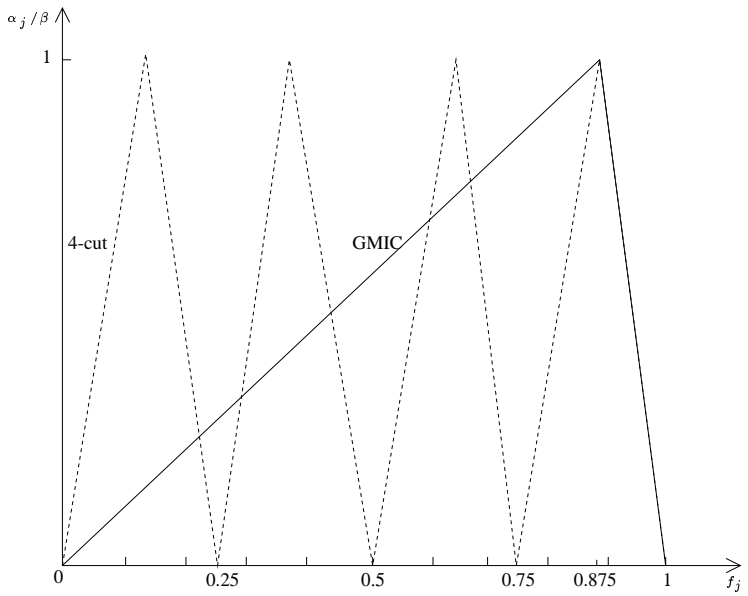

Figure 7: GMIC and 4-cut when $f_0 = 0.875$

f_j value			k-cut	GMIC
$(\frac{i}{k}, \frac{i+1}{k}]$, $0 \le i \le l-1$	$(\frac{i}{k}, f_0 - \frac{l-i}{k}]$	$(\frac{i}{k}, \frac{i}{l} f_0)$	1	2
		$\frac{i}{l} f_0$	1	1
		$(\frac{i}{l} f_0, f_0 - \frac{l-i}{k}]$	2	1
	$(f_0 - \frac{l-i}{k}, \frac{i+1}{k}]$	$(f_0 - \frac{l-i}{k}, \frac{i+1}{l+1} f_0]$	2	1
		$\frac{i+1}{l+1} f_0$	1	1
		$(\frac{i+1}{l+1} f_0, \frac{i+1}{k}]$	1	2
$(\frac{l}{k}, \frac{l+1}{k}]$	$(\frac{l}{k}, f_0)$		1	2
	f_0		1	1
	$(f_0, \frac{l+1}{k}]$		1	2
$(\frac{i}{k}, \frac{i+1}{k}]$, $l+1 \le i \le k-1$	$(\frac{i}{k}, f_0 - \frac{l-i}{k}]$	$(\frac{i}{k}, \frac{k-i}{k-l} f_0 + \frac{i-l}{k-l})$	1	2
		$\frac{k-i}{k-l} f_0 + \frac{i-l}{k-l}$	1	1
		$(\frac{k-i}{k-l} f_0 + \frac{i-l}{k-l}, f_0 - \frac{l-i}{k}]$	2	1
	$(f_0 - \frac{l-i}{k}, \frac{i+1}{k}]$	$(f_0 - \frac{l-i}{k}, \frac{k-i-1}{k-l-1} f_0 + \frac{i-l}{k-l-1})$	2	1
		$\frac{k-i-1}{k-l-1} f_0 + \frac{i-l}{k-l-1}$	1	1
		$(\frac{k-i-1}{k-l-1} f_0 + \frac{i-l}{k-l-1}, \frac{i+1}{k}]$	1	2

Table 2: comparison between k-cut and GMIC when $0 < l < kf_0 < l+1 < k$

f_j value			k-cut	GMIC
$(\frac{i}{k}, \frac{i+1}{k}]$, $0 \le i \le k-2$	$(\frac{i}{k}, f_0 - \frac{k-i-1}{k}]$	$(\frac{i}{k}, \frac{i}{k-1} f_0)$	1	2
		$\frac{i}{k-1} f_0$	1	1
		$(\frac{i}{k-1} f_0, f_0 - \frac{k-i-1}{k}]$	2	1
	$(f_0 - \frac{k-i-1}{k}, \frac{i+1}{k}]$	$(f_0 - \frac{k-i-1}{k}, \frac{i+1}{k} f_0]$	2	1
		$\frac{i+1}{k} f_0$	1	1
		$(\frac{i+1}{k} f_0, \frac{i+1}{k}]$	1	2
$(\frac{k-1}{k}, 1]$	$(\frac{k-1}{k}, f_0)$		1	2
	f_0		1	1
	$(f_0, 1]$		1	1

Table 3: comparison between k-cut and GMIC when $0 < l < kf_0 < l+1 = k$

The observations in Figure 6 and Figure 7 can be quantitatively summarized in Table 2 and Table 3 respectively in the general cases $0 < l < kf_0 < l+1 < k$ and $0 < l < kf_0 < l+1 = k$ for some integer l. Again we use the numbers 1 and 2 in the tables to rank the strength of the cuts on x_j in a decreasing order. Note that the case listed as $f_j = 1$ in Table 2 and

Table 3 corresponds to the actual case $f_j = 0$. Now Theorem 3 follows from observing the figures and tables.

THEOREM 3. *Assume that f_i is uniformly distributed in $[0, 1)$. Let $k \geq \lceil \frac{1}{f_0} \rceil$ and kf_0 be non-integral.*

(i) If $k > \lceil kf_0 \rceil$, then the probability that the k-cut generated from the tableau row is strictly stronger on x_i than the GMIC generated from the row is $\frac{1}{2}$, which is equal to the probability that the GMIC generated from the row is strictly stronger on x_i than the k-cut generated from the row.

(ii) If $k = \lceil kf_0 \rceil$, then the probability that the k-cut generated from the tableau row is strictly stronger on x_i than the GMIC generated from the row is $\frac{f_0}{2}$, which is equal to the probability that the GMIC generated from the row is strictly stronger on x_i than the k-cut generated from the row. □

In the following, we compare the GMIC to a group of k-cuts when $f_0 \geq \frac{1}{2}$.

THEOREM 4. *Assume $f_0 \geq \frac{1}{2}$ and $q < mf_0 < q + 1$ for some integer $q \geq 1$ and integer $m \geq 2$. Then*

(i) When $f_j \in (0, \frac{f_0}{q+1}]$, the GMIC is strictly stronger on x_j than any k-cut $(2 \leq k \leq m)$.

(ii) When $f_j \in (\frac{f_0}{q+1}, f_0]$, there exists some k-cut $(2 \leq k \leq m)$ such that the k-cut is stronger on x_j than the GMIC, and the "strictly stronger" property holds for all $f_j \in (\frac{f_0}{q+1}, f_0]$ except a finite number of isolated points.

(iii) If $f_0 > \frac{1}{2}$, then the 2-cut and the GMIC are equally strong on x_j when $f_j \in [f_0, 1)$. If $f_0 = \frac{1}{2}$, then the 3-cut is strictly stronger on x_j than the GMIC when $f_j \in (\frac{1}{2}, \frac{3}{4})$.

PROOF. (i) In the case of $i = 0$ in Table 2 and Table 3, the GMIC is strictly stronger on x_j than any k-cut $(2 \leq k \leq m)$ when $f_j \in (0, \frac{f_0}{q+1}]$.

(ii) Let k' be the largest integer between 2 and m such that $\lceil k'f_0 \rceil = k'$. First, let us look at Table 2 for the comparison, where we only consider the integer value of k in $[k' + 1, m]$. We decrease the value of k from m to $k' + 1$ in a way to ensure that the integer l decreases one at a time. Each time when l has a lower value, we choose $i = 0$ and 1 to identify two sub-intervals, in which the k-cut is strictly stronger on x_j than the GMIC. If we connect together all these sub-intervals, then we get an interval $(\frac{f_0}{q+1}, \frac{f_0}{k'-1}]$. From how we construct this interval, we know that, when $f_j \in (\frac{f_0}{q+1}, \frac{f_0}{k'-1}]$, there exists some k-cut $(k' + 1 \leq k \leq m)$ such that the k-cut is stronger on x_j than the GMIC, and the "strictly stronger" property holds for all $f_j \in (\frac{f_0}{q+1}, \frac{f_0}{k'-1}]$ except a few isolated points.

Now let us compare the k-cuts with GMIC in Table 3, assuming that the integer value of k is taken in $[2, k']$. Similarly as how we identify and connect

the sub-intervals above, we decrease the value of k once a time, meanwhile the value of l decreases once a time as well. Each time we choose $i = 0$ and 1 to figure out two sub-intervals, in which the k-cut is strictly stronger on x_j than the GMIC. Connecting together all these sub-intervals, we obtain an interval $(\frac{f_0}{k'}, \frac{1}{2}]$. Also we see in Table 3 that 2-cut is strictly stronger on x_j than the GMIC when $f_j \in (\frac{1}{2}, f_0)$. Therefore, when $f_j \in (\frac{f_0}{k'}, f_0]$, there exists some k-cut $(2 \leq k \leq k')$ such that the k-cut is stronger on x_j than the GMIC, and the "strictly stronger" property holds for all $f_j \in (\frac{f_0}{k'}, f_0]$ except a few isolated points.

(iii) In the case of $f_0 > \frac{1}{2}$, Table 3 tells us that the 2-cut and the GMIC are equally strong on x_j when $f_j \in [f_0, 1)$.

In the case of $f_0 = \frac{1}{2}$, the 3-cut is strictly stronger on x_j than the GMIC when $f_j \in (\frac{1}{2}, \frac{3}{4})$, which can be seen if we look at the case of $k = 3$, $l = 1$ and $i = 2$ in Table 2. □

4. Computational results

In this section, we test computationally how the k-cuts compare to GMIC for 0-1 and bounded knapsack problems and for integer programs with multiple constraints.

The knapsack problems have the following form:

$$\begin{aligned} \text{Max} \quad & \textstyle\sum_{j=1}^{n} p_j x_j \\ \text{s.t.} \quad & \textstyle\sum_{j=1}^{n} w_j x_j \leq c, \\ 0 \leq x_j \leq b_j & \text{ and integer, } \ 1 \leq j \leq n, \end{aligned}$$

where p_j, w_j and c are positive numbers and b_j is a positive integer. The 0-1 knapsack problems have $b_j = 1$ for all $1 \leq j \leq n$.

For the 0-1 knapsack problems, we generate both p_j and w_j randomly uniformly in [1,1000] and $c = 0.5 \sum_{j=1}^{n} w_j$, following Martello and Toth [19]. When we generate the bounded knapsack problems, p_j and w_j are set to be uniformly random in [1,1000], the values b_j are uniformly random in [5,10], and c is set to $0.5 \sum_{j=1}^{n} b_j w_j$. The number n of variables ranges from 10 to 10000, which represents the size of the problem.

In Table 4 and Table 5, we compare the gap closed after adding one k-cut to the LP relaxation of the knapsack problems, where k ranges from 1 to 5. The percentage figures in the tables represent the gap between the LP optimum and the IP optimum closed by adding the k-cuts. The asterisks '$*$' in the upper right corner of some percentage values indicate those values that are the largest among all the percentage values in their row. These two tables show that, for both the 0-1 and the bounded cases, each k-cut has

the same chance of being the best cut, over all values of k. The GMICs do not show any extraordinary advantage over other k-cuts. This fact is in accordance with the results in Theorem 1 and Theorem 3 in Section 3.

Problem set	Number of variables	gap closed by 1-cut	gap closed by 2-cut	gap closed by 3-cut	gap closed by 4-cut	gap closed by 5-cut
problem 1	10	76.31%	100.00%*	56.79%	92.22%	32.39%
problem 2	10	93.71%*	56.88%	29.82%	17.64%	6.18%
problem 3	100	5.94%	23.46%*	8.32%	3.50%	3.33%
problem 4	100	30.29%*	14.87%	15.66%	28.72%	23.21%
problem 5	1000	12.25%	6.06%	5.20%	25.74%	32.18%*
problem 6	1000	13.15%	14.19%*	9.51%	0.22%	5.20%
problem 7	5000	40.74%	56.48%*	34.26%	46.30%	55.56%
problem 8	5000	9.01%	9.91%	8.11%	13.51%*	11.71%
problem 9	10000	2.94%	4.41%	29.41%*	5.88%	5.88%
problem 10	10000	7.22%	21.65%	8.25%	7.22%	27.84%*

Table 4: individual k-cuts in 0-1 knapsack problems

Problem set	Number of variables	gap closed by 1-cut	gap closed by 2-cut	gap closed by 3-cut	gap closed by 4-cut	gap closed by 5-cut
problem 1	10	84.30%	70.17%	100.00%*	60.92%	52.18%
problem 2	10	19.73%	34.12%*	21.51%	32.19%	6.48%
problem 3	100	13.94%	13.94%	21.91%	29.51%	61.08*
problem 4	100	55.67%*	49.90%	27.29%	42.70%	43.18%
problem 5	1000	18.49%*	18.49%*	18.01%	18.49%*	18.49%*
problem 6	1000	8.74%*	4.11%	2.57%	1.80%	3.34%
problem 7	5000	8.26%	18.18%	14.05%	22.31%*	14.88%
problem 8	5000	1.42%	0.71%	14.18%*	0.71%	0.71%
problem 9	10000	9.19%	5.95%	5.41%	6.49%	11.89%*
problem 10	10000	2.68%	11.41%*	4.03%	8.05%	2.01%

Table 5: individual k-cuts in bounded knapsack problems

Problem set	Number of variables	gap closed by 1-cut	gap closed by $[1, 10]$-cuts	gap closed by $[1, 50]$-cuts
problem 1	10	53.98%	92.57%	100%
problem 2	10	94.05%	100%	100%
problem 3	50	19.19%	37.58%	40.24%
problem 4	50	8.11 %	35.01%	35.64%
problem 5	100	30.78%	77.04%	86.18%
problem 6	100	53.09%	100%	100%
problem 7	500	18.39%	25.06%	52.18%
problem 8	500	16.39%	49.76%	55.06%
problem 9	1000	6.79%	43.82%	54.58%
problem 10	1000	92.16%	100%	100%
problem 11	5000	15.77%	45.00%	57.31%
problem 12	5000	15.27%	39.69%	74.05%
problem 13	10000	3.70%	17.78%	24.44%
problem 14	10000	6.35%	14.81%	23.28%

Table 6: accumulated k-cuts in 0-1 knapsack problems

Problem set	Number of variables	gap closed by 1-cut	gap closed by $[1, 10]$-cuts	gap closed by $[1, 50]$-cuts
problem 1	10	66.95%	100%	100%
problem 2	10	74.29%	100%	100%
problem 3	50	41.87%	64.95%	77.30%
problem 4	50	6.45%	66.51%	70.99%
problem 5	100	39.62%	66.00%	69.04%
problem 6	100	20.38%	58.16%	94.71%
problem 7	500	15.25%	43.17%	71.29%
problem 8	500	2.22%	2.22%	66.37%
problem 9	1000	6.30%	20.82%	39.45%
problem 10	1000	6.46%	39.38%	81.88%
problem 11	5000	8.62%	17.24%	25.00%
problem 12	5000	11.56%	37.41%	41.50%
problem 13	10000	2.80%	6.54%	17.76%
problem 14	10000	1.67%	16.67%	36.67%

Table 7: accumulated k-cuts in bounded knapsack problems

Problems with 500 0-1 var.	Number of constraints	gap closed by 1-cut	gap closed by $[1, 10]$-cuts	gap closed by $[1, 50]$-cuts
problem 1	5 constr.	12.53%	15.56%	15.62%
problem 2	10 constr.	1.13%	2.27%	2.57%
problem 3	50 constr.	4.04%	4.16%	4.16%
problem 4	100 constr.	2.32%	2.32%	2.32%
probelms with 500 bd var.	Number of constraints	gap closed by 1-cut	gap closed by $[1, 10]$-cuts	gap closed by $[1, 50]$-cuts
problem 5	5 constr.	2.42%	5.33%	5.46%
problem 6	10 constr.	5.09%	5.09%	5.09%
problem 7	50 constr.	0.75%	0.79%	0.79%
problem 8	100 constr.	1.88%	1.92%	1.92%

Table 8: accumulated k-cuts in integer programming problems

In Table 6 and Table 7, we compare the cuts cumulatively: the third column corresponds to the case of just adding the GMIC, the next column corresponds to adding all the k-cuts with $1 \leq k \leq 10$, and the last column corresponds to adding all the k-cuts with $1 \leq k \leq 50$. The data imply that adding several k-cuts can close the gap effectively and works better than any single k-cut. In some cases, the improvement is very significant. The results in Theorem 2 and Theorem 4 are partly reflected in these tables.

In Table 8 we consider applying k-cuts to the 0-1 and bounded integer programming problems with more than one constraint. Each problem has 500 variables and from 5 to 100 constraints. The objective function and each constraint are randomly generated as in the problems in Tables 4-7. We generate k-cuts from all the rows of the tableau in which the basic variable has a fractional value. The results of Table 8 show that the effect of the k-cuts tails off dramatically as the number of constraints increases. This

is rather disappointing. Even more disappointing is the observation that adding groups of k-cuts provides only marginal improvements over adding the GMICs alone. We do not have any good theoretical explanation for this.

Obviously, the problems with more than one constraint are more meaningful from a practical point of view. However, in this case, adding the k-cuts for all $k \leq 50$ is not efficient as clearly illustrated in Table 8. A further investigation of k-cuts might still be interesting, especially for problems with general integer variables, since there are not many powerful cuts available for the general integer case. To investigate the usefulness of using k-cuts in practice, one should solve problems to optimality, say in combination with the branch-and-bound method. Our purpose in this section was much more limited. We only wanted to illustrate the theorems of Section 3.

5. Extension to the mixed case

The mixed integer programming (MIP) problems have the standard form

$$\begin{aligned} &\text{Min } cx + dy \\ \text{s.t. } &Ax + By = b, \\ &x \in Z^n_+,\ y \in R^p_+, \end{aligned}$$

where A and B are rational matrices, and c, d and b are rational vectors.

Solving the LP relaxation of the MIP gives tableau rows of the form

$$x_0 = \bar{a}_0 + \textstyle\sum_{j \in J} \bar{a}_j(-x_j) + \sum_{j \in S} \bar{g}_j(-y_j), \tag{5}$$

where $f_0 = \bar{a}_0 - \lfloor \bar{a}_0 \rfloor > 0$, J and S denote the sets of the non-basic integer variables and the non-basic continuous variables.

Using the notation from Section 2, the GMIC is, in the mixed case:

$$\sum_{j: f_j \leq f_0} f_j x_j + \frac{f_0}{1-f_0} \sum_{j: f_j > f_0} (1-f_j)x_j + \sum_{j: g_j \geq 0} g_j y_j - \frac{f_0}{1-f_0} \sum_{j: g_j < 0} g_j y_j \geq f_0. \tag{6}$$

Note that the difference between the GMICs in (3) and (6) is that the cut in (6) contains terms for the continuous variables.

Applying the same proof procedure as that for Lemma 1 shows that the following cutting plane is valid for MIP:

$$\sum_{j: \tilde{f}_j \leq \tilde{f}_0} \tilde{f}_j x_j + \frac{\tilde{f}_0}{1-\tilde{f}_0} \sum_{j: \tilde{f}_j > \tilde{f}_0} (1-\tilde{f}_j)x_j + \sum_{j: g_j \geq 0} k g_j y_j - \frac{\tilde{f}_0}{1-\tilde{f}_0} \sum_{j: g_j < 0} k g_j y_j \geq \tilde{f}_0, \tag{7}$$

which is called the *k-cut in the mixed case*, or the k-cut in brief.

Comparing the cuts in (6) and (7), we can see that, in the mixed case, the k-cut ($k \geq 2$) is weaker in the continuous variables than the GMIC. Similar to the picture introduced in Section 3 for an integer variable, one can construct a picture for a continuous variable y_j. Let $\beta = \tilde{f}_0$ and $\alpha_j = kg_j$ when $g_j \geq 0$, $\frac{\tilde{f}_0}{1-\tilde{f}_0} kg_j$ when $g_j < 0$. Then the function α_j/β is a linear function of $|g_j|$ with nonnegative slope and the slope increases as k increases, which weakens the cut on y_j. Hence it seems logical that increasing k too much leads to bad inequalities.

We observed in practice that k-cuts ($k \geq 2$) in the mixed case are often not as good as the GMICs. So there exists an intrinsic difficulty in extending the results of the pure IP case to the MIP case. One might still be able to use them conditionally. For example, in the case where $f_0 < \frac{1}{2}$ and $g_j \geq 0$ for all j (or $f_0 > \frac{1}{2}$ and $g_j \leq 0$ for all j), the results of Theorem 1 and Theorem 2 can be applied to obtain k-cuts ($2 \leq k \leq \lfloor \frac{1}{f_0} \rfloor$) (or $2 \leq k \leq \lfloor \frac{1}{1-f_0} \rfloor$) that are as powerful as the GMIC and provide a variety of cutting planes.

The following two tables illustrate the deterioration of k-cuts in the mixed case as k increases. In the knapsack problems, the slack variables are considered to be continuous. The corresponding GMIC and k-cuts are generated in the forms of (6) and (7). The quality of the cuts is measured by the gap closed after adding one cut. We can see from Table 9 and Table 10 that the gap closed tends to decrease as k increases, which implies that the k-cut deteriorates as k becomes larger.

problem set	gap closed by 1-cut	gap closed by 10-cut	gap closed by 100-cut	gap closed by 1000-cut	gap closed by 10000-cut
problem 1	10.50%	13.22%	3.17%	0.87%	0.06%
problem 2	5.94%	2.47%	1.89%	0.77%	0.03%
problem 3	5.60%	0.27%	7.96%	0.27%	0.02%
problem 4	30.31%	24.19%	5.83%	0.09%	0.09%
problem 5	51.93%	16.76%	6.86%	0.55%	0.08%

Table 9: individual k-cuts in 0-1 knapsack problems

problem set	gap closed by 1-cut	gap closed by 10-cut	gap closed by 100-cut	gap closed by 1000-cut	gap closed by 10000-cut
problem 1	39.62%	25.72%	5.32%	0.43%	0.03%
problem 2	19.69%	44.52%	11.26%	3.52%	0.38%
problem 3	11.78%	44.48%	7.58%	0.40%	0.00%
problem 4	13.02%	0.32%	2.34%	0.98%	0.06%
problem 5	42.30%	8.46%	18.93%	2.11%	0.15%

Table 10: individual k-cuts in bounded knapsack problems

A further extension of GMIC combines several rows from the LP tableau with integer multipliers. This idea has attracted attention recently [1, 4, 17, 24]. This topic is currently under further investigation.

Acknowledgments

We would like to thank the referees for their helpful comments. This work was supported in part by NSF grant DMI-0098427 and ONR grant N00014-97-1-0196.

References

[1] Andersen, K., G. Cornuéjols, Y. Li. 2001. Improving cuts in integer programming. INFORMS Annual Conference, Miami Beach, (http://www.informs.org/Conf/Miami2001//TALKS/TC11.html).

[2] Balas, E., S. Ceria, G. Cornuéjols, N. Natraj. 1996. Gomory cuts revisited. *Operations Research Letters* **19** 1–9.

[3] Bowman Jr., V., G. Nemhauser. 1970. A finite proof for modified Dantzig cuts in integer programming. *Naval Research Logistics Quarterly* **17** 309–313.

[4] Ceria, S., G. Cornuéjols, M. Dawande. 1995. Optimizing generalized Gomory cuts for pure and mixed integer programs. *Lecture Notes in Computer Science 920*. E. Balas and J. Clausen eds., Integer Programming and Combinatorial Optimization.

[5] Chvátal, V. 1973. Edmonds polytope and a hierarchy of combinatorial problems. *Discrete Mathematics* **4** 305–337.

[6] Dawande, M., private communication.

[7] Garfinkel, R., G. Nemhauser. 1972. *Integer Programming*. New York, Wiley.

[8] Glover, F. 1966. Generalized cuts in Diophantine programming. *Management Science* **13** 254–268.

[9] Gomory, R. 1958. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society* **64** 275–278.

[10] Gomory, R. 1960. Solving linear programming problems in integers. *Combinatorial Analysis*. R.E. Bellman and M. Hall, Jr., eds., American Mathematical Society, 211–216.

[11] Gomory, R. 1963. An algorithm for integer solutions to linear programs. *Recent Advances in Mathematical Programming*. R. Graves and P. Wolfs, eds., McGraw-Hill, 269–302.

[12] Gomory, R. 1969. Some polyhedra related to combinatorial problems. *Linear Algebra and Its Applications* **2** 451–558.

[13] Gomory, R. 1999. Corner polyhedra and their connection with cutting planes. Integer Programming Symposium, T.J. Watson Research Center.

[14] Gomory, R., A. Hoffman. 1963. On the convergence of an integer programming process. *Naval Research Logistics Quarterly* **10** 121–124.

[15] Gomory, R., E. Johnson. 1972. Some continuous functions related to corner polyhedra, Part I. *Mathematical Programming* **3** 23–85.

[16] Gomory, R., E. Johnson. 1972. Some continuous functions related to corner polyhedra, Part II. *Mathematical Programming* **3** 359–389.

[17] Günlük, O., Y. Pochet. 2001. Mixing mixed-integer inequalities. *Mathematical Programming A*. **90** 429–457.

[18] Johnson, E. 1974. On the group problem for mixed integer programming. *Mathematical Programming Study* **2** 137–179.

[19] Martello, S., P. Toth. 1989. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons.

[20] Martin, G. 1963. An accelerated Euclidean algorithm for integer linear programming. *Recent Advances in Mathematical Programming*. R. Graves and P. Wolfs, eds., McGraw-Hill.

[21] Miliotis, P. 1978. Using cutting planes to solve the symmetric traveling salesman problem. *Mathematical Programming* **15** 177–188.

[22] Schrijver, A. 1986. *Theory of Linear and Integer Programming*. John Wiley & Sons.

[23] Vizvári, B. 1989. Two algorithms to get strong Gomory cuts. *Optimization* **20** 117–126.

[24] Marchand, H., L. Wolsey. 2001. Aggregation and mixed integer rounding to solve MIPs. *Operations Research* **49** 363–371.

Accepted by Jan Karel Lenstra; received January 2001; revised December 2001, May 2002; accepted May 2002.