

AN EXAMPLE OF THE GOMORY CUTTING PLANE ALGORITHM

Consider the integer programme

$$\max z = 3x_1 + 4x_2$$

subject to

$$\begin{aligned} 3x_1 - x_2 &\leq 12 \\ 3x_1 + 11x_2 &\leq 66 \\ x &\in \mathbb{N}^2 \end{aligned}$$

The first linear programming relaxation is

$$\max z = 3x_1 + 4x_2$$

subject to

$$\begin{aligned} 3x_1 - x_2 &\leq 12 \\ 3x_1 + 11x_2 &\leq 66 \\ x &\geq 0 \end{aligned}$$

After introducing slackness variables s_1 and s_2 , we obtain the simplex tableau

z	x_1	x_2	s_1	s_2	rhs	BV
1	-3	-4	0	0	0	$z = 0$
0	3	-1	1	0	12	$s_1 = 12$
0	3	11	0	1	66	$s_2 = 66$

We use MAPLE's `linalg` package to take care of the simplex steps:

```
> with(linalg):
> A := matrix(3,6,[1,-3,-4,0,0,0,0,3,-1,1,0,12,0,3,11,0,1,66]);
      [1  -3  -4  0  0  0]
      [
A := [0  3  -1  1  0  12]
      [
      [0  3  11  0  1  66]
> A := mulrow(A,3,1/11); # x2 enters, s2 leaves
      [1  -3  -4  0  0  0]
      [
A := [0  3  -1  1  0  12]
      [
      [ 3           1   ]
      [0  --   1  0  --  6]
      [ 11          11  ]
> A := pivot(A,3,3);
      [ -21          4   ]
      [1  ---  0  0  --  24]
      [ 11           11  ]
```

```

      [
      [ 36      1      ]
A := [0  --  0  1  -- 18]
      [ 11      11      ]
      [
      [ 3      1      ]
      [0  --  1  0  --  6]
      [ 11      11      ]
> A := mulrow(A,2,11/36); x1 enters, s1 leaves
      [ -21      4      ]
      [1  ---  0  0  -- 24]
      [ 11      11      ]
      [
      [      11  1  11]
A := [0  1  0  --  --  --]
      [      36 36  2 ]
      [
      [ 3      1      ]
      [0  --  1  0  --  6]
      [ 11      11      ]
> A := pivot(A,2,2);
      [      7  5  69]
      [1  0  0  --  --  --]
      [      12 12  2 ]
      [
      [      11  1  11]
A := [0  1  0  --  --  --]
      [      36 36  2 ]
      [
      [      -1  1  9]
      [0  0  1  --  --  -]
      [      12 12  2]

```

So we have found the solution of the first LPR, namely $x_1 = 11/2$ and $x_2 = 9/2$. This solution is non-integral, so we seek a cut. For this purpose, we choose a row of the optimal tableau *with a non-integral right-hand side*. For instance, the second row of the optimal tableau says

$$x_1 = \frac{11}{2} - \frac{11}{36}s_1 - \frac{1}{36}s_2 = 5 + \frac{1}{2} - \frac{11}{36}s_1 - \frac{1}{36}s_2.$$

We can express this as

$$(C) \quad x_1 - 5 = \frac{1}{2} - \frac{11}{36}s_1 - \frac{1}{36}s_2.$$

We argue that the inequality

$$(G) \quad \frac{1}{2} - \frac{11}{36}s_1 - \frac{1}{36}s_2 \leq 0$$

is a cut. Indeed, it is a valid inequality for, if x and s are integral, then it follows from Equation (C) that

$$\frac{1}{2} - \frac{11}{36}s_1 - \frac{1}{36}s_2 \in \mathbb{Z}.$$

Any integer-feasible s is also non-negative, and so

$$\frac{1}{2} - \frac{11}{36}s_1 - \frac{1}{36}s_2 \leq 1/2.$$

The integrality of the left-hand side then implies that Equation (G) holds. To show that Equation (G) is a cut, there remains to show that there exists a vector (x, s) that is feasible for the current relaxation, but that violates Equation (G). The optimal solution of the relaxation is one such vector, since it is such that $s = 0$.

This argument is easily generalised. Suppose that the current LP relaxation has an optimal tableau with a row with a non-integral right-hand side r ; we write the corresponding as

$$x_{bv} = r - \sum_{x_j \in NBV} a_j x_j.$$

For any real number t , we write

$$[t] := \{n \in \mathbb{Z} : n \leq t\} \quad \text{and} \quad \{x\} := t - [t] \in [0, 1).$$

Then

$$t = [t] + \{t\}$$

and we can rewrite the equation for the row as

$$x_{bv} - [r] + \sum_{x_j \in NBV} [a_j]x_j = \{x\} - \sum_{x_j \in NBV} \{a_j\}x_j.$$

Then the inequality

$$\{x\} - \sum_{x_j \in NBV} \{a_j\}x_j \leq 0$$

is a *Gomory cut*.

Returning to our example, we introduce a new slack variable s_3 and rewrite the cut as

$$-\frac{11}{36}s_1 - \frac{1}{36}s_2 + s_3 = -\frac{1}{2}.$$

With this new variable and this new constraint, the simplex tableau becomes

```
> A := extend(A,1,1,0); # Gomory cut: 1/2-11/36*s1 -1/36*s2 <= 0
[
  7  5  69 ]
[1  0  0  -- -- -- 0]
[
  12  12  2  ]
[
  ]
[
  11  1  11 ]
[0  1  0  -- -- -- 0]
A := [
  36  36  2  ]
[
  ]
[
  -1  1  9  ]
[0  0  1  -- -- - 0]
[
  12  12  2  ]
[
  ]
[0  0  0  0  0  0 0]
> for i from 1 to 4 do A[i,7] := A[i,6] : A[i,6] := 0 : od :
A[4,4] := -11/36 :
A[4,5] := -1/36 : A[4,6] := 1 : A[4,7] := -1/2 :
print(A);
```

```

[      7  5  69]
[1  0  0  -- -- 0  --]
[      12 12  2 ]
[      ]
[      11  1  11]
[0  1  0  -- -- 0  --]
[      36 36  2 ]
[      ]
[      -1  1  9]
[0  0  1  -- -- 0  -]
[      12 12  2]
[      ]
[      -11 -1 -1]
[0  0  0  --- -- 1  --]
[      36 36  2 ]

```

The basic solution corresponding to this tableau is not feasible, since the right-hand side in the last row is negative. On the other hand, the coefficients in the first row are all non-negative— indicating dual-feasibility. So we use the *dual simplex method* to solve the relaxation.

```

> A := mulrow(A,4,-36/11);
[      7  5  69]
[1  0  0  -- -- 0  --]
[      12 12  2 ]
[      ]
[      11  1  11]
[0  1  0  -- -- 0  --]
[      36 36  2 ]
A := [      ]
[      -1  1  9]
[0  0  1  -- -- 0  -]
[      12 12  2]
[      ]
[      1 -36 18]
[0  0  0  1  -- --- --]
[      11 11  11]

> A := pivot(A,4,4);
[      4  21 369]
[1  0  0  0  -- -- ---]
[      11 11 11 ]
[      ]
[0  1  0  0  0  1  5]
[      ]
A := [      1 -3 51]
[0  0  1  0  -- -- ---]
[      11 11 11]
[      ]
[      1 -36 18]
[0  0  0  1  -- --- --]

```

$$[\quad \quad \quad 11 \quad 11 \quad 11]$$

This is optimal and LP-feasible, but not integral. For the next Gomory cut, we use the third row:

$$x_2 = \frac{51}{11} - \frac{1}{11}s_2 + \frac{3}{11}s_3.$$

So the cut is

$$\frac{7}{11} - \frac{1}{11}s_2 - \frac{8}{11}s_3 \leq 0.$$

We introduce a new slackness variable s_4 and a new constraint

$$-\frac{1}{11}s_2 - \frac{8}{11}s_3 + s_4 = -\frac{7}{11}.$$

> A := extend(A,1,1,0);

$$A := \begin{bmatrix} [& & & & 4 & 21 & 369 &] \\ [1 & 0 & 0 & 0 & -- & -- & --- & 0] \\ [& & & & 11 & 11 & 11 &] \\ [& & & & & & &] \\ [0 & 1 & 0 & 0 & 0 & 1 & 5 & 0] \\ [& & & & & & &] \\ [& & & & 1 & -3 & 51 &] \\ [0 & 0 & 1 & 0 & -- & -- & -- & 0] \\ [& & & & 11 & 11 & 11 &] \\ [& & & & & & &] \\ [& & & & 1 & -36 & 18 &] \\ [0 & 0 & 0 & 1 & -- & --- & -- & 0] \\ [& & & & 11 & 11 & 11 &] \\ [& & & & & & &] \\ [0 & 0 & 0 & 0 & 0 & 0 & 0 & 0] \end{bmatrix}$$

> for i from 1 to 5 do A[i,8] := A[i,7] : A[i,7] := 0 : od :
 A[5,8] := -7/11 : A[5,7] := 1 : A[5,6] := -8/11 :
 A[5,5] := -1/11 : print(A);

$$\begin{bmatrix} [& & & & 4 & 21 & 369] \\ [1 & 0 & 0 & 0 & -- & -- & 0 & ---] \\ [& & & & 11 & 11 & 11 &] \\ [& & & & & & &] \\ [0 & 1 & 0 & 0 & 0 & 1 & 0 & 5] \\ [& & & & & & &] \\ [& & & & 1 & -3 & 51] \\ [0 & 0 & 1 & 0 & -- & -- & 0 & --] \\ [& & & & 11 & 11 & 11] \\ [& & & & & & &] \\ [& & & & 1 & -36 & 18] \\ [0 & 0 & 0 & 1 & -- & --- & 0 & --] \\ [& & & & 11 & 11 & 11] \\ [& & & & & & &] \\ [& & & & -1 & -8 & -7] \\ [0 & 0 & 0 & 0 & -- & -- & 1 & --] \\ [& & & & 11 & 11 & 11] \end{bmatrix}$$

One step of the dual simplex method gives

```

> A := mulrow(A,5,-11/8);
      [
      [1  0  0  0  4  21  369]
      [  0  0  0  0  --  --  0  ---]
      [  0  0  0  0  11  11  11 ]
      [  0  0  0  0  0  1  0  5]
      [  0  0  0  0  1  -3  51]
      [0  0  1  0  --  --  0  --]
      A := [  0  0  0  0  11  11  11]
      [  0  0  0  0  1  -36  18]
      [0  0  0  1  --  ---  0  --]
      [  0  0  0  0  11  11  11]
      [  0  0  0  0  1  -11  7]
      [0  0  0  0  -  1  ---  -]
      [  0  0  0  0  8  8  8]

> A := pivot(A,5,6);
      [
      [1  0  0  0  1  21  255]
      [  0  0  0  0  -  0  --  ---]
      [  0  0  0  0  8  8  8 ]
      [  0  0  0  0  -1  11  33]
      [0  1  0  0  --  0  --  --]
      [  0  0  0  0  8  8  8 ]
      [  0  0  0  0  1  -3  39]
      A := [0  0  1  0  -  0  --  --]
      [  0  0  0  0  8  8  8 ]
      [  0  0  0  0  1  -9  9]
      [0  0  0  1  -  0  --  -]
      [  0  0  0  0  2  2  2]
      [  0  0  0  0  1  -11  7]
      [0  0  0  0  -  1  ---  -]
      [  0  0  0  0  8  8  8]

```

This is optimal, but not integral. For our next cut, we choose the penultimate row:

$$s_1 = \frac{9}{2} - \frac{1}{2}s_2 + \frac{9}{2}s_4.$$

This gives the Gomory cut

$$\frac{1}{2} - \frac{1}{2}s_2 - \frac{1}{2}s_4 \leq 0.$$

We introduce a new slackness variable s_5 and a new constraint

$$-\frac{1}{2}s_2 - \frac{1}{2}s_4 + s_5 = -\frac{1}{2}.$$

Thus

```

> A := extend(A,1,1,0);
      [
      [1  0  0  0  -  0  --  ---  0]
      [
      [
      [
      [0  1  0  0  --  0  --  --  0]
      [
      [
      [0  0  1  0  -  0  --  --  0]
A := [
      [
      [
      [0  0  0  1  -  0  --  -  0]
      [
      [
      [0  0  0  0  -  1  ---  -  0]
      [
      [
      [0  0  0  0  0  0  0  0  0]
> for i from 1 to 6 do A[i,9] := A[i,8] : A[i,8] := 0 : od :
A[6,9] := -1/2 : A[6,8] := 1 : A[6,7] := -1/2 : A[6,5] := -1/2 :
print(A);
      [
      [1  0  0  0  -  0  --  0  ---]
      [
      [
      [
      [0  1  0  0  --  0  --  0  --]
      [
      [
      [0  0  1  0  -  0  --  0  --]
      [
      [
      [0  0  0  1  -  0  --  0  -]
      [
      [
      [
      [0  0  0  0  -  1  ---  0  -]
      [
      [
      [
      [0  0  0  0  --  0  --  1  --]

```

[2 2 2]

One step of the dual simplex algorithm gives

```

> A := mulrow(A,6,-2);
      [
      [1  0  0  0  -  0  --  0  ---]
      [
      [
      [
      [0  1  0  0  --  0  --  0  --]
      [
      [
      [
      [0  0  1  0  -  0  --  0  --]
      A := [
      [
      [
      [0  0  0  1  -  0  --  0  -]
      [
      [
      [
      [0  0  0  0  -  1  ---  0  -]
      [
      [
      [0  0  0  0  1  0  1  -2  1]
> A := pivot(A,6,5);
      [
      [1  0  0  0  0  0  5  1  127]
      [
      [
      [
      [0  1  0  0  0  0  -  --  --]
      [
      [
      [
      A := [0  0  1  0  0  0  --  -  --]
      [
      [
      [0  0  0  1  0  0  -5  1  4]
      [
      [
      [0  0  0  0  0  1  --  -  -]
      [
      [
      [0  0  0  0  1  0  1  -2  1]

```

This is optimal, but still not integral! For our next cut, we take the second row:

$$x_1 = \frac{17}{4} - \frac{3}{2}s_4 + \frac{1}{4}s_5.$$

This gives the Gomory cut

$$\frac{1}{4} - \frac{1}{2}s_4 - \frac{3}{4}s_5 \leq 0.$$

We introduce a new slackness variable s_6 and write our new constraint as

$$-\frac{1}{2}s_4 - \frac{3}{4}s_5 + s_6 = -\frac{1}{4}.$$

The new tableau is then

```
> A := extend(A,1,1,0);
      [
      [1  0  0  0  0  0  5  1  127  ]
      [  -  -  ---  0]
      [  2  4  4  ]
      [  ]
      [  3  -1  17  ]
      [0  1  0  0  0  0  -  --  --  0]
      [  2  4  4  ]
      [  ]
      [  -1  1  19  ]
      [0  0  1  0  0  0  --  -  --  0]
A := [  2  4  4  ]
      [  ]
      [0  0  0  1  0  0  -5  1  4  0]
      [  ]
      [  -3  1  3  ]
      [0  0  0  0  0  1  --  -  -  0]
      [  2  4  4  ]
      [  ]
      [0  0  0  0  1  0  1  -2  1  0]
      [  ]
      [0  0  0  0  0  0  0  0  0  0]
> for i from 1 to 7 do A[i,10] := A[i,9] : A[i,9] := 0 : od :
A[7,10] := -1/4 :
A[7,9] := 1 : A[7,8] := -3/4 : A[7,7] := -1/2 : print(A);
      [
      [1  0  0  0  0  0  5  1  127]
      [  -  -  0  ---]
      [  2  4  4  ]
      [  ]
      [  3  -1  17]
      [0  1  0  0  0  0  -  --  0  --]
      [  2  4  4  ]
      [  ]
      [  -1  1  19]
      [0  0  1  0  0  0  --  -  0  --]
      [  2  4  4  ]
      [  ]
      [0  0  0  1  0  0  -5  1  0  4]
      [  ]
      [  -3  1  3]
      [0  0  0  0  0  1  --  -  0  -]
```

```

[           2   4   4]
[           ]
[0  0  0  0  1  0  1 -2  0  1]
[           ]
[           -1 -3  -1]
[0  0  0  0  0  0  -- --  1  --]
[           2   4   4 ]

```

One step of the dual simplex algorithm then gives

```

> A := mulrow(A,7,-4/3);
[           5   1  127]
[1  0  0  0  0  0  - -  0  ---]
[           2   4   4 ]
[           ]
[           3  -1  17]
[0  1  0  0  0  0  - --  0  --]
[           2   4   4 ]
[           ]
[           -1  1  19]
[0  0  1  0  0  0  -- -  0  --]
[           2   4   4 ]
A := [           ]
[0  0  0  1  0  0  -5  1  0  4]
[           ]
[           -3  1   3]
[0  0  0  0  0  1  -- -  0  -]
[           2   4   4 ]
[           ]
[0  0  0  0  1  0  1 -2  0  1]
[           ]
[           2  -4  1]
[0  0  0  0  0  0  -  1  --  -]
[           3   3  3]
> A := pivot(A,7,8);
[           7   1  95]
[1  0  0  0  0  0  -  0  -  --]
[           3   3  3 ]
[           ]
[           5  -1  13]
[0  1  0  0  0  0  -  0  --  --]
[           3   3  3 ]
[           ]
[           -2   1  14]
[0  0  1  0  0  0  --  0  -  --]
[           3   3  3 ]
[           ]
[           -17  4  11]
A := [0  0  0  1  0  0  ---  0  -  --]
[           3   3  3 ]

```

$$\begin{array}{cccccccc}
 [& & & & & & &] \\
 [& & & & & -5 & 1 & 2] \\
 [0 & 0 & 0 & 0 & 0 & 1 & -- & 0 & - & -] \\
 [& & & & & 3 & 3 & 3] \\
 [& & & & & & &] \\
 [& & & & & 7 & -8 & 5] \\
 [0 & 0 & 0 & 0 & 1 & 0 & - & 0 & -- & -] \\
 [& & & & & 3 & 3 & 3] \\
 [& & & & & & &] \\
 [& & & & & 2 & -4 & 1] \\
 [0 & 0 & 0 & 0 & 0 & 0 & - & 1 & -- & -] \\
 [& & & & & 3 & 3 & 3]
 \end{array}$$

Optimal, but not integral. We take the second row for our next cut. After introducing the new slackness variable s_7 , we write the new constraint as

$$-\frac{2}{3}s_4 - \frac{2}{3}s_6 + s_7 = -\frac{1}{3}.$$

Then, after the dual simplex step,

```
> A := pivot(A,8,9);
```

$$\begin{array}{cccccccc}
 [& & & & & & & 1 & 63] \\
 [1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & - & --] \\
 [& & & & & & & & & 2 & 2] \\
 [& & & & & & & & &] \\
 [& & & & & & & & & -1 & 9] \\
 [0 & 1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & -- & -] \\
 [& & & & & & & & & 2 & 2] \\
 [& & & & & & & & &] \\
 [& & & & & & & & & 1 & 9] \\
 [0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & - & -] \\
 [& & & & & & & & & 2 & 2] \\
 [& & & & & & & & &] \\
 A := [0 & 0 & 0 & 1 & 0 & 0 & -7 & 0 & 0 & 2 & 3] \\
 [& & & & & & & & &] \\
 [& & & & & & & & & 1 & 1] \\
 [0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 0 & 0 & - & -] \\
 [& & & & & & & & & 2 & 2] \\
 [& & & & & & & & &] \\
 [0 & 0 & 0 & 0 & 1 & 0 & 5 & 0 & 0 & -4 & 3] \\
 [& & & & & & & & &] \\
 [0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 0 & -2 & 1] \\
 [& & & & & & & & &] \\
 [& & & & & & & & & -3 & 1] \\
 [0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -- & -] \\
 [& & & & & & & & & 2 & 2]
 \end{array}$$

Not integral! We use the second row for the next cut:

$$-\frac{1}{2}s_7 + s_8 = -\frac{1}{2}.$$

The optimal tableau for the new problem is then

```

> A := pivot(A,9,10);
      [1  0  0  0  0  0  2  0  0  0  1  31]
      [
      [0  1  0  0  0  0  2  0  0  0 -1  5]
      [
      [0  0  1  0  0  0 -1  0  0  0  1  4]
      [
      [0  0  0  1  0  0 -7  0  0  0  4  1]
      [
A := [0  0  0  0  0  1 -2  0  0  0  1  0]
      [
      [0  0  0  0  1  0  5  0  0  0 -8  7]
      [
      [0  0  0  0  0  0  2  1  0  0 -4  3]
      [
      [0  0  0  0  0  0  1  0  1  0 -3  2]
      [
      [0  0  0  0  0  0  0  0  0  1 -2  1]

```

This optimal— and integral. The solution of our IP is thus

$$x_1 = 5 \quad \text{and} \quad x_2 = 4.$$