The tableau below gives the LP solution to the relaxation of an integer programming problem with maximization objective. All $a_{ij}$ and $b_i$ coefficients in the original problem are integer.

<table>
<thead>
<tr>
<th>Row no.</th>
<th>Basic variables</th>
<th>Coefficients</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$z$</td>
<td>1 0.1 0.3 0 0.2 0 0</td>
<td>23.1</td>
</tr>
<tr>
<td>1</td>
<td>$x_6$</td>
<td>0 1.3 -0.3 0 -1.0 0 1</td>
<td>5.3</td>
</tr>
<tr>
<td>2</td>
<td>$x_3$</td>
<td>0 0 1.1 1 0.4 0 0</td>
<td>1.6</td>
</tr>
<tr>
<td>3</td>
<td>$x_5$</td>
<td>0 -0.8 -0.2 0 -0.5 1 0</td>
<td>3.7</td>
</tr>
</tbody>
</table>

a. Write out all Gomory cuts that can be derived from the tableau.

b. Write out the Dantzig cut that can be derived from the tableau.

c. Add the Gomory cut with the largest right-hand-side value to the tableau and use the dual simplex method to find the new solution. Perform these computations by hand.

---

a. Gomory cuts that can be derived

\[0.3x_1 + 0.7x_2 \geq 0.3\]

\[0.1x_2 + 0.4x_4 \geq 0.6\]

\[0.2x_1 + 0.8x_2 + 0.5x_4 \geq 0.7\]

From objective function: \[0.1x_1 + 0.3x_2 + 0.2x_2 \geq 0.1\]

b. Write out the Dantzig cut that can be derived from the tableau.

\[x_1 + x_2 + x_4 \geq 1\]

c. Add the Gomory cut with the largest right-hand-side value to the tableau and use the dual simplex method to find the new solution. Perform these computations by hand.
Let $x_7$ leave the basis. $x_2$ will enter.

This is the new optimal solution after the cut was added. Note, that the new objective function value should be $z = 22.8375$.

---

21. (Generalized Gomory cut) Suppose we wish to use a cutting plane technique to solve a pure integer program. Assume that the LP relaxation has been solved and the $i$th constraint in the simplex tableau is

$$x_{B(i)} + \sum_{j \in Q} \tilde{a}_{ij}x_j = \tilde{b}_i$$

where $x_{B(i)}$ is the $i$th basic variable and $Q$ is the set of nonbasic variables. Multiply this equation by any rational number $h$. Now, using the same approach outlined in Section 8.4 to derive the basic Gomory fractional cut given in Eq. (14), derive from the equation above a more generalized version of Eq. (14).

Consider the linear problem

$$\text{Maximize } \{c \mathbf{x} : \mathbf{x} \in S\}$$

where $S = \{\mathbf{x} \in \mathbb{R}_+^n : \mathbf{A} \mathbf{x} \leq \mathbf{b}\}$ be the set of feasible points. Suppose that we have a representation of $S$ given by
\[ x_{B(i)} = \tilde{b}_i - \sum_{j \in Q} \tilde{a}_{ij} x_j, \quad i = 0, 1, \ldots, m \quad (S1) \]

so that the basic solution to determined by (S1) is \( x_{B(i)} = \tilde{b}_i, i = 0, \ldots, m, x_j = 0 \) for \( j \in Q \). If we multiple (S1) by \( h \neq 0 \), we get

\[ hx_{B(i)} + \sum_{j \in Q} h\tilde{a}_{ij} x_j = \tilde{b}_i \]

The nonnegative requirement \( x_j \geq 0 \) for all \( j \) implies that

\[ \lfloor hx_{B(i)} \rfloor + \sum_{j \in Q} \lfloor h\tilde{a}_{ij} \rfloor x_j \leq h\tilde{b}_i \quad (S2) \]

and because \( x \) must be integer, the left-hand side of (S2) must also be an integer. This means that the left-hand side of (S2) cannot exceed the integer part of the right-hand side, implying that

\[ \lfloor hx_{B(i)} \rfloor + \sum_{j \in Q} \lfloor h\tilde{a}_{ij} \rfloor x_j \leq \lfloor h\tilde{b}_i \rfloor \quad (S3) \]

Multiplying (S1) by \( \lfloor h \rfloor \) and then subtracting (S3) yields

\[ \sum_{j \in Q} \left( \lfloor h \rfloor \tilde{a}_{ij} - \lfloor h\tilde{a}_{ij} \rfloor \right) x_j \geq \lfloor h \rfloor \tilde{b}_i - \lfloor h\tilde{b}_i \rfloor \quad (S3) \]

which is called the fundamental cut. By using (S3) in a variety of ways, several different cuts and associated algorithms can be developed.
29. Solve the problem below with branch and bound. Perform several iterations by hand and then use the accompanying Excel add-ins.

Maximize \[ 2x_1 + 3x_2 + 4x_3 + 7x_4 \]

subject to \[ 4x_1 + 6x_2 - 2x_3 + 8x_4 = 20 \]
\[ x_1 + 2x_2 - 6x_3 + 7x_4 = 10 \]
\[ x_j \geq 0 \text{ and integer}, \ j = 1, 2, 3, 4 \]

The complete branch and bound solution is below. It was obtained with the Teach IP add-in.

<table>
<thead>
<tr>
<th>Node</th>
<th>Level</th>
<th>Variable</th>
<th>Value</th>
<th>Up/Down</th>
<th>Visit</th>
<th>Relax</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>27.6471</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>-1</td>
<td>1</td>
<td>Infeasible</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>-1</td>
<td>2</td>
<td>21.3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>21.2308</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>Infeasible</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>-1</td>
<td>2</td>
<td>21</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>2</td>
<td>17.8824</td>
</tr>
</tbody>
</table>

Branch X(3) up at 2
Infeasible: Backtrack Level 1: Branch X(3) down at 1
Branch X(1) down at 1
Branch X(2) down at 0
Infeasible: Backtrack Level 3: Branch X(2) up at 1
Integer: Replace incumbent: Backtrack Level 3: Branch X(1) up at 2
Fathom: Backtrack Level 2 1 0
:Finished