Introduction

In all the previous lectures in linear programming discussed so far, the design variables considered are supposed to take any real value. However, in practical problems like minimization of labor needed in a project, it makes little sense in assigning a value like 5.6 to the number of labourers. In situations like this, one natural idea for obtaining an integer solution is to ignore the integer constraints and use any of the techniques previously discussed and then round-off the solution to the nearest integer value. However, there are several fundamental problems in using this approach:

1. The rounded-off solutions may not be feasible.
2. The objective function value given by the rounded-off solutions (even if some are feasible) may not be the optimal one.
3. Even if some of the rounded-off solutions are optimal, checking all the rounded-off solutions is computationally expensive (\(2^n\) possible round-off values to be considered for an \(n\) variable problem)

Types of Integer Programming

When all the variables in an optimization problem are restricted to take only integer values, it is called an all-integer programming problem. When the variables are restricted to take only discrete values, the problem is called a discrete programming problem. When only some variable values are restricted to take integer or discrete, it is called mixed integer or discrete programming problem. When the variables are constrained to take values of either zero or 1, then the problem is called zero-one programming problem.

Integer Linear Programming

Integer Linear Programming (ILP) is an extension of linear programming, with an additional restriction that the variables should be integer valued. The standard form of an ILP is of the form,
\[
\text{max } c^T X \\
\text{subject to } AX \leq b \\
X \geq 0
\]

\(X\) must be integer valued

The associated linear program dropping the integer restrictions is called \textit{linear relaxation LR}. Thus, LR is less constrained than ILP. If the objective function coefficients are integer, then for minimization, the optimal objective for ILP is greater than or equal to the rounded-off value of the optimal objective for LR. For maximization, the optimal objective for ILP is less than or equal to the rounded-off value of the optimal objective for LR.

For a minimization ILP, the optimal objective value for LR is less than or equal to the optimal objective for ILP and for a maximization ILP, the optimal objective value for LR is greater than or equal to that of ILP. If LR is infeasible, then ILP is also infeasible. Also, if LR is optimized by integer variables, then that solution is feasible and optimal for IP.

A most popular method used for solving all-integer and mixed-integer linear programming problems is the cutting plane method by Gomory (Gomory, 1957).

\textbf{Gomory’s Cutting Plane Method for All – Integer Programming}

Consider the following optimization problem.

\[
\begin{align*}
\text{Maximize} & \quad Z = 3x_1 + x_2 \\
\text{subject to} & \quad 2x_1 - x_2 \leq 6 \\
& \quad 3x_1 + 9x_2 \leq 45 \\
& \quad x_1, x_2 \geq 0 \\
\end{align*}
\]

\(x_1\) and \(x_2\) are integers

The graphical solution for the \textit{linear relaxation} of this problem is shown below.
It can be seen that the solution is $x_1 = 4\frac{5}{7}, x_2 = 3\frac{3}{7}$ and the optimal value of $Z = 17\frac{4}{7}$.

The feasible solutions accounting the integer constraints are shown by red dots. These points are called integer lattice points. The original feasible region is reduced to a new feasible region by including some additional constraints such that an extreme point of the new feasible region becomes an optimal solution after accounting for the integer constraints.

The graphical solution for the example previously discussed taking $x_1$ and $x_2$ as integers are shown below. Two additional constraints (MN and OP) are included so that the original feasible region ABCD is reduced to a new feasible region AEFGCD. Thus the solution for this ILP is $x_1 = 4$, $x_2 = 3$ and the optimal value is $Z = 15$. 
Gomory proposed a systematic method to develop these additional constraints known as *Gomory constraints*. 

**Generation of Gomory Constraints:**

Let the final tableau of an LP problem consist of $n$ basic variables (original variables) and $m$ non basic variables (slack variables) as shown in the table below. The basic variables are represented as $x_i \ (i=1,2,...,n)$ and the non basic variables are represented as $y_j \ (j=1,2,...,m)$. 

---

D Nagesh Kumar, IISc, Bangalore
Table 1

<table>
<thead>
<tr>
<th>Basis</th>
<th>Z</th>
<th>Variables</th>
<th>$b_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$x_1$  $x_2$ ... $x_i$  ... $x_n$ $y_1$ $y_2$ ... $y_j$ ... $y_m$</td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>1</td>
<td>0     0           0            0      $c_1$ $c_2$ $c_j$ $c_m$ $b$</td>
<td></td>
</tr>
<tr>
<td>$x_1$</td>
<td>0</td>
<td>1     0           0            0      $c_{11}$ $c_{12}$ $c_{1j}$ $c_{1m}$ $b_1$</td>
<td></td>
</tr>
<tr>
<td>$x_2$</td>
<td>0</td>
<td>0     1           0            0      $c_{21}$ $c_{22}$ $c_{2j}$ $c_{2m}$ $b_2$</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>$x_i$</td>
<td>0</td>
<td>0     0           1            0      $c_{31}$ $c_{32}$ $c_{3j}$ $c_{3m}$ $b_i$</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>$x_n$</td>
<td>0</td>
<td>0     0           0            1      $c_{41}$ $c_{42}$ $c_{4j}$ $c_{4m}$ $b_n$</td>
<td></td>
</tr>
</tbody>
</table>

Choose any basic variable $x_i$ with the highest fractional value. If there is a tie between two basic variables, arbitrarily choose any of them as $x_i$. Then from the $i^{th}$ equation of table,

$$x_i = b_i - \sum_{j=1}^{m} c_{ij} y_j$$ ........(1)

Express both $b_i$ and $c_{ij}$ as an integer value plus a fractional part.

$$b_i = \overline{b}_i + \beta_i$$ ............(2)

$$c_{ij} = \overline{c}_{ij} + \alpha_{ij}$$ ............(3)

where $\overline{b}_i$, $\overline{c}_{ij}$ denote the integer part and $\beta_i$, $\alpha_{ij}$ denote the fractional part. $\beta_i$ will be a strictly positive fraction ($0 < \beta_i < 1$) and $\alpha_{ij}$ is a non-negative fraction ($0 \leq \alpha_{ij} < 1$).

Substituting equations (2) and (3) in (1), equation (1) can be written as

$$\beta_i - \sum_{j=1}^{m} \alpha_{ij} y_j = x_i - \overline{b}_i - \sum_{j=1}^{m} \overline{c}_{ij} y_j$$ ............(4)

For all the variables $x_i$ and $y_j$ to be integers, the right hand side of equation (4) should be an integer.

$$\beta_i - \sum_{j=1}^{m} \alpha_{ij} y_j = \text{integer}$$ ............(5)
Since \( \alpha_{ij} \) are non-negative integers and \( y_j \) are non-negative integers, the term \( \sum_{j=1}^{m} \alpha_{ij} y_j \) will always be a non-negative number. Thus we have,

\[
\left( \beta_i - \sum_{j=1}^{m} \alpha_{ij} y_j \right) \leq \beta_i < 1 \quad \text{……(6)}
\]

Hence the constraint can be expressed as

\[
\beta_i - \sum_{j=1}^{m} \alpha_{ij} y_j \leq 0 \quad \text{……(7)}
\]

By introducing a slack variable \( s_i \) (which should also be an integer), the Gomory constraint can be written as

\[
s_i - \sum_{j=1}^{m} \alpha_{ij} y_j = -\beta_i \quad \text{……(8)}
\]

**General procedure for solving ILP:**

1. Solve the given problem as an ordinary LP problem neglecting the integer constraints. If the optimum values of the variables are integers itself, then there is nothing more to be done.

2. If any of the basic variables has fractional values, introduce the Gomory constraints as discussed in the previous section. Insert a new row with the coefficients of this constraint, to the final tableau of the ordinary LP problem (Table 1).

3. Solve this by applying the dual simplex method.

   Since the value of \( y_j = 0 \) in Table 1, the Gomory constraint equation becomes

   \[
s_i = -\beta_i \text{ which is a negative value and thus infeasible. Dual simplex method is used to obtain a new optimal solution that satisfies the Gomory constraint.}
\]

4. Check whether the new solution is all-integer or not. If all values are not integers, then a new Gomory constraint is developed from the new simplex tableau and the dual simplex method is applied again.

5. This process is continued until an optimal integer solution is obtained or it shows that the problem has no feasible integer solution.
Thus, the fundamental idea behind cutting planes is to add constraints to a linear program until the optimal basic feasible solution takes on integer values. Gomory cuts have the property that they can be generated for any integer program, but has the disadvantage that the number of constraints generated can be enormous depending upon the number of variables.