Integer Linear Programs


**Integer Linear Programs**

Consider the following version of Woody’s problem:

\[
\begin{align*}
x_1 &= \text{chairs per day}, \quad x_2 = \text{tables per day} \\
\text{max } z &= 35x_1 + 60x_2 \quad \text{profits} \\
14x_1 + 26x_2 &\leq 190 \quad \text{pine} \\
15x_2 &\leq 60 \quad \text{mahogany} \\
8x_1 + 3x_2 &\leq 92 \quad \text{labor} \\
x_1 &\geq 0 \quad x_2 \geq 0
\end{align*}
\]

Suppose that Woody insists on making a whole number of items each day. This is an example of an integer linear program (ILP), where the objective function and constraints satisfy the proportionality and additivity assumptions, but the variables are required in addition to be integer, that is, they violate the continuity condition. **Mixed integer LPs** are those where some of the variables are required to be integer while others can be continuously valued.
Graphical Picture of Woody’s IP
Dealing with Integer LPs

We wish to solve an ILP (in max inequality form):

\[
\begin{align*}
\text{max } z &= c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \\
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &\leq b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &\leq b_2 \\
\vdots & \quad \vdots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &\leq b_m \\
x_1 \geq 0, \quad x_2 \geq 0, \quad \ldots \quad x_n \geq 0 \\
x_1, \ldots, x_n &\text{ integer}
\end{align*}
\]

Main Tool: Try to solve this problem as a LP, by dropping the integrality requirement. The corresponding LP is called the LP relaxation \((R)\) to \((I)\).

Facts:

1. The optimal solution value for \((R)\) provides an upper bound on that of \((I)\).
   (Proof: The set of feasible points for \((R)\) is a superset of those of \((I)\).)

2. If \((R)\) has an optimal solution that is integer, then this is also the optimal solution to \((I)\).
   (Proof: Any integer solution to \((I)\) provides a lower bound on the objective function value.)
Example: Woody’s original IP (2- and 3-variable versions) have optimal solutions that are integer ($x_1 = 12$ and $x_2 = 2$), and hence optimal to the corresponding ILP.

Unfortunately, this version of the LP has optimal solution $x_1 = 10.98$ and $x_2 = 1.4$ with objective function value 468.01. We know that the optimal integer solution has value no greater than 468.01 (in fact, no greater than 468).

Question: How can we tell in advance that the relaxed LP will have an integer optimal solution? Are there classes of LPs that are guaranteed to always produce integer optimal solutions?

Answer: Yes! The main class comprises those LPs associated with network flow problems.

network matrix: Any matrix $A$ whose columns have the property that their nonzero entries consist of at most one +1 and at most one -1.
Example: Shortest Path Problem

Consider finding the shortest path from $s = 1$ to $t = 4$ in the following network:

The LP for this problem can be thought of as sending one unit of flow through the system starting at $s$ and ending at $t$, that is,

$$\min 2x_{12} + 6x_{13} + 3x_{23} + 5x_{32} + 8x_{24} + 4x_{34}$$

where $x_{ij} = \begin{cases} 1 & \text{if arc } (i, j) \text{ is on the path} \\ 0 & \text{otherwise} \end{cases}$
Lemma: Let $A$ be any $m \times m$ network matrix. Then $A$ is unimodular, that is $\det A = +1, -1$ or 0.

Proof: Recall the definition of determinant:

$$
\det A = \begin{cases} 
a_{11}, & m = 1 \\
\sum_{i=1}^{m} (-1)^{i+j} a_{ij} \det A_{[i,j]}, & m > 1
\end{cases}
$$

where $A_{[i,j]}$ is the matrix obtained by deleting the $i^{th}$ row and $j^{th}$ column, and the sum can be taken over any column $j$. Clearly if $m = 1$ then $A = [\pm1, 0]$, and so has determinant $\pm1, 0$. Now proceed by induction, and suppose $m > 1$. Since $A$ is a network matrix, then every column of $A$ can have at most one $+1$ and one $-1$. If there is a column $j$ with only one nonzero element $a_{kj}$, then

$$
\det A = (-1)^{k+j} a_{kj} A_{[k,j]} = (\pm1)(\pm1) \det A_{[k,j]}
$$

which is $\pm1, 0$ by induction, since $A_{[k,j]}$ is an $(m-1) \times (m-1)$ network matrix. If every column has two nonzero elements, then adding the rows of $A$ together gives the zero vector, and so $A$ is singular, and hence has determinant 0.
**Theorem:** Let \((P)\) be an LP whose associated matrix is a **network matrix**, and whose right-hand side coefficients are **integer**. Then all basic solutions to \((P)\) are **integer**.

**Proof:** We will consider the constraints of \((P)\) to be in *equality* form \(Ax = b\), since any general LP can be put into equality form by adding *slack* columns, and the resulting matrix will continue to be a network matrix. We will also assume that \(A\) has full row rank \(m\), since \(A\) will continue to be a network matrix after the deletion of any redundant rows. Then any basic solution is of the form \(\hat{x} = [\hat{x}_B, \hat{x}_n]\), where \(B\) is a nonsingular submatrix of columns of \(A\) and \(\hat{x}_B = \bar{b} = B^{-1}b\), \(\hat{x}_N = 0\), is partitioned corresponding to \(B\). Then recalling Cramer’s Rule, we write the coefficients of \(x_B\) as

\[
x_{B_i} = \frac{\det B|_i}{\det B}, \quad i = 1, \ldots, m
\]

where \(B|_i\) is the matrix obtained from \(B\) by substituting \(b\) for the \(i^{th}\) column of \(B\).

Now the **numerators** of these fractions are always integer as long as \(b\) is integer. The **denominator**, moreover, is always \(\pm 1\) by the lemma. Thus \(x_B\) will always be integer.
**Corollary:** For any LP $(P)$ whose associated matrix is a network matrix, if $(P)$ is feasible then it has an integer feasible solution, and if $(P)$ has an optimal solution then it has an integer optimal solution.

**Proof:** Recall that if $(P)$ has a feasible solution, then it always has a basic feasible solution, and if $(P)$ has an optimal solution, then it always has a basic feasible optimal solution.

**Comments:**

1. The integrality conclusions of the theorem continue to hold for the dual solution to a network LP, so long as the costs are likewise integer.
2. Actually constructing the solutions to network LPs can be done much more easily than for general LPs, and in fact, there are specialized LP solvers for network LPs that perform several times faster than applying a general LP solver.
3. The network matrices above are “almost the entire” class of matrices that have this integrality property, in a technical sense we will not go into here.
Dealing with Noninteger LP Solutions

What if the LP does not produce an integer solution?

Rounding Heuristic: Try rounding the noninteger values up or down to the nearest feasible integer point.

duality gap: Difference between the current best integer solution and the optimal relaxation LP solution. If the duality gap is 0 — or < 1 if costs are integer — then the current best solution is optimal.

Example: For our Woody’s example, round the current solution \( x_1 = 10.98 \) and \( x_2 = 1.4 \) down to \( x_1 = 10, \ x_2 = 1 \ (z = 410) \), or if you are more clever, to \( x_1 = 11, \ x_2 = 1 \ (z = 445) \). This has a duality gap of \( 468 - 445 = 23 \).

The optimal ILP solution, surprisingly, is \( x_1 = 8, \ x_2 = 3 \ (z = 460) \), still with a duality gap of 8.

Rounding is a good heuristic when the integer solutions have large values, and there are no equality constraints. When the problem has a restricted feasible region, or the variables are binary (0-1), it does not perform well. Here we need more sophisticated techniques for determining optimality.
The Branch-and-Bound Technique for Solving ILPs

When the relaxation fails to produce an integer optimum, we need to restrict the LP to cut out the noninteger points from the feasible region. This is done by branching on a variable with noninteger value. In particular, suppose $\hat{x}$ is the current optimal LP relaxation solution, with $\hat{x}_j$ noninteger. Produce the two branching subproblems, by bounding $x_j$ away from its noninteger value:

$$\begin{align*}
\text{max } z &= cx \\
Ax &\leq b \\
(I^-) &\quad x \geq 0 \text{ integer} \\
x_j &\leq \lfloor \hat{x}_j \rfloor \\
\text{max } z &= cx \\
Ax &\leq b \\
(I^+) &\quad x \geq 0 \text{ integer} \\
x_j &\geq \lceil \hat{x}_j \rceil
\end{align*}$$

where $\lfloor a \rfloor$ is the greatest integer less than or equal to $a$ and $\lceil a \rceil$ is the least integer greater than or equal to $a$.

**Fact:** All integer solutions to $(I)$ will be feasible to one of $(I^+)$ and $(I^-)$. Thus the optimal solution to $(I)$ can be found by solving $(I^+)$ and $(I^-)$ and choosing the solution with the larger objective function value.
The Branch-and-Bound Tree

Solving each of the subproblems can in turn result in more branching, and many problems must be considered. To keep track of these problems, we use a branch-and-bound tree. Each node of this tree consists of one of the subproblems to be solved, with the root of the tree the original problem. At each node the relaxation LP is solved, and a branching is performed on a chosen noninteger optimal variable value, creating two new nodes corresponding to the two subproblems given above.

fathoming: A node is fathomed if it is determined that no more branching needs to be done at that node.
Fathoming Criteria

• **fathom by infeasibility:** If the relaxation LP has no feasible solutions, then it will have no integer feasible solutions, and so no more work needs to be done on that subproblem.

• **fathom by integrality:** If the optimal relaxation LP solution is integer, then it is clearly the optimal solution to that subproblem, and in fact a feasible — called candidate — solution to the original ILP. Again, no more work needs to be done on that subproblem.

• **fathom by nonoptimality:** Whenever a fathom by integrality occurs, then the candidate solution is noted, and the best of the candidate solutions for the entire problem is retained. If at a particular node, the current relaxation LP solution has objective function value less than or equal to the objective function value of the current best candidate solution, then no integer solution for this subproblem will be better than the current best candidate solution (since the LP objective function value is an upper bound on the optimal integer solution value for that subproblem), and again no more work needs to be done on that problem.
Example

Starting with the ILP

$$\max z = 35x_1 + 60x_2$$
$$14x_1 + 26x_2 \leq 190$$

$$(S_1)$$
$$15x_2 \leq 60$$
$$8x_1 + 3x_2 \leq 92$$
$$x_1 \geq 0 \quad x_2 \geq 0$$

$$x_1, x_2 \text{ integer}$$

We have the optimal solution $\hat{x}_1 = 10.98$ and $\hat{x}_2 = 1.4$ with objective function value 468.01. This is not integer, so we branch on a noninteger variable, say $x_2$, to form the two LPs

$$\max z = 35x_1 + 60x_2$$
$$14x_1 + 26x_2 \leq 190$$

$$(S_2)$$
$$15x_2 \leq 60$$
$$8x_1 + 3x_2 \leq 92$$
$$x_2 \leq 1$$
$$x_1 \geq 0 \quad x_2 \geq 0$$

$$x_1, x_2 \text{ integer}$$

$$\max z = 35x_1 + 60x_2$$
$$14x_1 + 26x_2 \leq 190$$

$$(S_3)$$
$$15x_2 \leq 60$$
$$8x_1 + 3x_2 \leq 92$$
$$x_2 \geq 2$$
$$x_1 \geq 0 \quad x_2 \geq 0$$

$$x_1, x_2 \text{ integer}$$

Select node $(S_3)$ for consideration. The optimal solution for $(S_3)$ is $\hat{x}_1 = 9.86$, $\hat{x}_2 = 2$, with objective value $z = 465$. This is also not integer, so we branch on noninteger variable $x_1$ to get LPs

$$\max z = 35x_1 + 60x_2$$
$$14x_1 + 26x_2 \leq 190$$

$$(S_4)$$
$$15x_2 \leq 60$$
$$8x_1 + 3x_2 \leq 92$$
$$x_2 \geq 2$$
$$x_1 \leq 9$$
$$x_1 \geq 0 \quad x_2 \geq 0$$

$$x_1, x_2 \text{ integer}$$

$$\max z = 35x_1 + 60x_2$$
$$14x_1 + 26x_2 \leq 190$$

$$(S_5)$$
$$15x_2 \leq 60$$
$$8x_1 + 3x_2 \leq 92$$
$$x_2 \geq 2$$
$$x_1 \geq 10$$
$$x_1 \geq 0 \quad x_2 \geq 0$$

$$x_1, x_2 \text{ integer}$$
Select node \((S_5)\) for consideration. The relaxed LP here is infeasible, and thus node \((S_5)\) is fathomed by infeasibility. We next select node \((S_4)\) for consideration. The optimal solution for \((S_4)\) is \(\hat{x}_1 = 9, \hat{x}_2 = 2.46\), with objective value \(z = 462.7\). This is also not integer, so we branch on noninteger variable \(x_2\) to get LPs

\[
\begin{align*}
\max z &= 35x_1 + 60x_2 \\
14x_1 + 26x_2 &\leq 190 \\
15x_2 &\leq 60 \\
8x_1 + 3x_2 &\leq 92 \\
x_2 &\leq 2 \\
x_1 &\leq 9 \\
x_1 \geq 0 & \quad x_2 \geq 0 \\
x_1, x_2 \text{ integer}
\end{align*}
\]

Select node \((S_6)\) for consideration. The optimal solution for \((S_6)\) is \(\hat{x}_1 = 9, \hat{x}_2 = 2\), with objective value \(z = 435\). This solution is integer, and so \((S_6)\) is fathomed by integrality, and the candidate solution is stored as the current best solution. We next select node \((S_7)\). The optimal solution for \((S_7)\) is \(\hat{x}_1 = 8, \hat{x}_2 = 3\), with objective value \(z = 460\). This is also integer, and its objective function value is larger than the current best solution found at node \((S_6)\). We therefore replace that solution with the current one as the best candidate solution, and \((S_7)\) is fathomed by integrality.

Finally, select node \((S_2)\). The optimal solution for \((S_2)\) is \(\hat{x}_1 = 11.13, \hat{x}_2 = 1\), with objective value \(z = 449.4\). This value is less than that of the current best candidate solution, and so node \((S_2)\) is fathomed by nonoptimality.

Since all nodes have now been fathomed, we have completed the evaluation of all relevant subproblems, and therefore the current best candidate solution, \(\hat{x}_1 = 8, \hat{x}_2 = 3, z = 460\), is the optimal solution to the ILP.
Cutting Planes

An “easy” way of obtaining ILP solutions:
Suppose we solve Woody’s original ILP to obtain optimal solution $\hat{x}_1 = 10.98$ and $\hat{x}_2 = 1.4$. What if, instead of proceeding with a complicated branch and bound routine, we simply add the constraint

$$2x_1 + 3x_2 \leq 25$$

to the ILP. This constraint has the property that it (1) cuts out the current noninteger solution, and (2) does not cut off any other integer points. Such a constraint is called a cutting plane for the ILP relaxation. It follows that adding a cutting plane insures that the optimal integer solution will continue to be that of the original ILP, and the optimal solution to the new relaxation solution will not be the same solution that we obtained in the previous LP. In this case, the the optimal relaxation solution $\hat{x}_1 = 8$, $\hat{x}_2 = 3$ is in fact integer, and hence it is immediately an optimal solution to Woody’s LP.

Cutting planes are extremely powerful tools in solving ILPs, since they can produce optimal solutions in a relatively small number of steps.
Questions

• How can we prove that a particular constraint is a cutting plane, in particular, that it does not cut out any integer solutions?

• How can we find cutting planes?

• How good is a particular cutting plane, in terms of insuring a quick solution to the ILP?
Rounding Inequalities

**Obvious fact:** For any $\leq$ inequality with **integer** left-hand-side coefficients and **noninteger** right-hand-side, the same inequality with the right-hand-side **rounded down** to the nearest integer will continue to be a valid inequality for the ILP.

**Application:** Suppose the first constraint for Woody’s ILP was 189 instead of 190. The optimal solution is $\hat{x}_1 = 10.994$, $\hat{x}_2 = 1.349$. If we divide the first constraint by 2, we get valid constraint

$$7x_1 + 13x_2 \leq 94.5$$

Rounding the r.h.s. down to 94 and multiplying by 2, we get stronger constraint

$$14x_1 + 26x_2 \leq 188$$

which cuts out the current optimal relaxation solution (although it still does not result in an optimal ILP solution).
Using Surrogate Constraints

Better Application: Woody’s actual ILP does not admit a simple rounding inequality like that above. Suppose, however, we consider the surrogate constraint obtained by adding $31 \times \text{(inequality#1)} + 8 \times \text{(inequality#3)}$:

$$498x_1 + 830x_2 \leq 6626.$$  

Dividing through by the common denominator 166 for the l.h.s. gives rounded inequality

$$3x_1 + 5x_2 \leq \lceil 39.916 \rceil = 39$$

which is now a valid constraint to the ILP. Adding this inequality to the set of inequalities results in an ILP whose relaxed LP has integer optimal solution $\hat{x}_1 = 8, \hat{x}_2 = 3$, which is now provably optimal to the ILP.
Finding Cutting Planes from Optimal LP Tableaus

It turns out that we can always find a surrogate constraint that will round to produce a valid cutting plane in the following way. Suppose we start with an integer program all of whose parameters are integer. We solve this using any LP method and determine the optimal LP relaxation solution together with the associated optimal tableau. In Woody’s IP problem, for example, the optimal tableau is:

<table>
<thead>
<tr>
<th>$z$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>rhs</th>
<th>bas.var.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>−.723</td>
<td>1</td>
<td>1.265</td>
<td>39.036</td>
<td>$s_2$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>.048</td>
<td>0</td>
<td>−.084</td>
<td>1.398</td>
<td>$x_2$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>−.018</td>
<td>0</td>
<td>.157</td>
<td>10.976</td>
<td>$x_1$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2.259</td>
<td>0</td>
<td>.422</td>
<td>468.0</td>
<td></td>
</tr>
</tbody>
</table>

Now the optimal solution $x_1 = 10.976$, $x_2 = 1.398$, and $x_3 = 0$ is not an all integer one, and so this is not a feasible solution to the integer program. Pick one of the fractional constraints (heuristically best: the one with the fraction closest to .5), and write the corresponding tableau equation down. In this case the second equation is chosen, namely,

$$x_2 + .048s_1 − .084s_3 = 1.398$$
Now **round down** each of the left-hand-side coefficients of this equation, and **subtract the remainder** from the opposite side of the equation, obtaining the **inequality**

\[ x_2 + 0s_1 - 1s_3 = 1.398 - .048s_1 - .916s_3 \leq 1.398 \]

since \( s_1 \) and \( s_3 \) must be nonnegative. Now since the LHS coefficients are integer, then the LHS value must be **integer** for any integer values of \( x_1, x_2, x_3 \) (and hence \( s_1, s_2, s_3 \), since all coefficients are integer). Thus we can **round down** the right-hand-side value to get the inequality

\[ x_2 - s_3 \leq 1 \]

or equivalently,

\[ 8x_1 + 4x_2 \leq 93 \]

**Fact:** This is a **valid** inequality, and it is **violated** by the current optimal solution. We call such a cut a **Gomory-Hu** cut associated with this tableau.
The Gomory-Hu Cutting Plane Algorithm

The Gomory-Hu Algorithm uses the fact that every non-integer optimal tableau can be used to add a valid cutting plane using the rounding method on one of the (surrogate) rows of the tableau.

Steps of Gomory-Hu Algorithm:

1. Solve the LP relaxation. If the optimal LP solution is integer, STOP, this solution is also optimal to the IP.

2. If the LP solution is not integer, then the optimal tableau has at least one row whose RHS value is noninteger. Choose one of these rows \( i \). (Heuristically best choice: one whose fractional part is closest to 1/2.)

3. Round down each of the LHS coefficients and the RHS value for Row \( i \) to obtain a Gomory-Hu inequality. Add this to the set of cutting planes already produced. Go back to Step 1.
Efficiency of the Gomory-Hu Cutting Plane Algorithm

By a careful choice of rounding cuts, the Gomory-Hu Algorithm can be shown to obtain an integer (hence optimal) solution after adding a finite number of cuts. For Woody’s example, it took 8 Gomory-Hu inequalities to obtain the optimal solution.

<table>
<thead>
<tr>
<th>Added Inequality</th>
<th>New Soln. $x_1$</th>
<th>$x_2$</th>
<th>Obj. Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. $8x_1 + 4x_2 \leq 93$</td>
<td>10.91</td>
<td>1.43</td>
<td>467.8</td>
</tr>
<tr>
<td>2. $6x_1 + 7x_2 \leq 75$</td>
<td>10.69</td>
<td>1.55</td>
<td>467.2</td>
</tr>
<tr>
<td>3. $6x_1 + 8x_2 \leq 76$</td>
<td>10.36</td>
<td>1.73</td>
<td>466.4</td>
</tr>
<tr>
<td>4. $15x_1 + 26x_2 \leq 200$</td>
<td>10.44</td>
<td>1.67</td>
<td>465.6</td>
</tr>
<tr>
<td>5. $16x_1 + 26x_2 \leq 210$</td>
<td>10.00</td>
<td>1.92</td>
<td>465.4</td>
</tr>
<tr>
<td>6. $11x_1 + 19x_2 \leq 146$</td>
<td>10.62</td>
<td>1.54</td>
<td>463.8</td>
</tr>
<tr>
<td>7. $6x_1 + 9x_2 \leq 77$</td>
<td>9.33</td>
<td>1.93</td>
<td>463.7</td>
</tr>
<tr>
<td>8. $2x_1 + 3x_2 \leq 25$</td>
<td>8.00</td>
<td>3.00</td>
<td>460</td>
</tr>
</tbody>
</table>

Unfortunately, this method in general must generate an extraordinarily large number of cuts to obtain an integer solution, and so is too inefficient to be useful.
Deep Cuts and Facet Cuts

The inefficiency of the Gomory-Hu method stems from the fact that the cuts produced are very weak, in terms of not cutting off enough of the noninteger region, or equivalently, not giving inequalities that are tight in any integer solution.

**Goal:** to develop “deep” cuts in the above sense.

**integer hull** of an ILP \((I)\): the convex hull \(C(I)\) of the set of integer solutions to \((I)\).

**Consequence of Weyl’s Theorem:** \(C(I)\) is a polytope, that is, it can be represented by a set of equalities and inequalities whose extreme points are a subset of the integer solutions to \((I)\).

**Corollary:** Any optimal bfs for the linear program

\[
(C) \quad \max cx \\
\text{s.t.} \quad x \in C(I)
\]

will be an optimal solution to \((I)\).

**Proof:** We know that any optimal bfs \(x^*\) for \((C)\) will be an extreme point for \(C(I)\), hence will be integer, and thus feasible to \((I)\). Further, since any optimal IP solution \(x^I\) is clearly in \(C(I)\), then the objective function value of \(x^*\) must be the same as that of \(x^I\), and so \(x^*\) is also optimal to \((I)\).
Example

The integer hull for Woody’s ILP looks like

with extreme integer solutions

\([0, 0], [0, 4], [6, 4], [8, 3], [11, 1], [11, 0]\),

and has the following minimal polytope description:

\[
14x_1 + 28x_2 \leq 196 \\
14x_1 + 21x_2 \leq 175 \\
15x_2 \leq 60 \\
8x_1 \leq 88 \\
x_1 \geq 0 \quad x_2 \geq 0
\]

That is, all six inequalities above define \textbf{facets} of the convex hull, and hence each is essential in solving the ILP for at least one pair of cost coefficients \([c_1, c_2]\).
On Finding Convex Hulls

Goal: Ideally, we would like to find a complete minimal description of $C(I)$, that is, a description all of whose inequalities define facets of $C(I)$.

Problems:

- There may be an exponentially large number of facets, in terms of the size of the description of the original ILP.

- Finding such a representation is a daunting, and sometimes provably difficult, task.

- Using such a representation, in terms of incorporating the cutting planes into the LP solver, may also be difficult.
A More Realistic Goal

Suppose we want to solve a particular class \( \mathcal{P} \) of ILPs of importance. That is, we want a \textbf{generic} method of attacking any particular instance \( I \in \mathcal{P} \). To do this using a cutting plane method, for any problem \( P \in \mathcal{P} \) we need to produce

(a) a linear program \( LP_I \) whose feasible region \textbf{contains} \( C(I) \), and provides a reasonably good approximation of \( C(I) \).

(b) a set \( K_I \) of potential \textbf{cutting planes}, each of which are satisfied by all points in \( C(I) \), and that further restrict the LP region for \( P \). These are not explicitly given (and in fact there may be exponentially many of these) but are produced when needed to force the LP to give an optimal solution to \( I \).
Properties of “Good” \( LP_I \) and \( \mathcal{K}_I \)

1. \( LP_I \) should give feasible solutions to \( I \) much of the time, and should also give optimal solutions quickly when cutting planes are added, or when branch-and-bound is performed. In particular, the equalities for \( LP_I \) should define the equality space for \( \mathcal{C}(I) \), and inequalities for \( LP_I \) should as much as possible be facets of \( \mathcal{C}(I) \).

2. An inequality in \( \mathcal{K}_I \) should form a high-dimensional face (preferably a facet) for \( \mathcal{C}(I) \).

3. Given any point \( \hat{x} \) feasible to \( LP_I \) but not in \( \mathcal{C}(I) \), there should be an efficient method of explicitly producing an inequality of \( K \in \mathcal{K}_I \) that is not satisfied by \( \hat{x} \).
A Generic Cutting Plane Heuristic

Suppose we wish to solve a particular instance $I$ of $\mathcal{P}$. We first produce $LP_I$ and identify the appropriate set $\mathcal{K}_I$ of inequalities that will be used in the algorithm.

Steps of the Cutting Plane Heuristic:

1. Solve $LP_I$. If the optimal LP solution $x^*$ is feasible to $I$ then STOP, $x^*$ is also optimal to $I$.

2. If $x^*$ is not feasible to $I$, then produce an inequality $K \in \mathcal{K}_I$ that is not satisfied by $x^*$. Add this inequality to $LP_I$ and go back to Step 1.

This is only a heuristic, in that unless $LP_I$ and $\mathcal{K}_I$ are strong enough to provide bounding inequalities for the optimal BFS, this algorithm may be forced to stop with no further inequalities from $\mathcal{K}_I$ being available to cut off the current LP solution.
The Branch-and-Cut Algorithm

The branch-and-cut method for solving integer programs combines both branch-and-bound and cutting planes. For given instance $I$ of $\mathcal{P}$, we again produce sets $LP_I$ and $\mathcal{K}_I$, and then proceed as follows:

1. Solve $LP_I$. If the optimal solution $x^*$ is feasible to $I$, STOP, $x^*$ is optimal to $I$.

2. Perform branch-and-bound on the problem for some amount of time to see if an optimal or near-optimal solution can be found. If a satisfactory solution is found, STOP.

3. If it does not appear that the branch-and-bound method is converging to an integer solution, then produce one or more violating inequalities from $\mathcal{K}_I$ to add to the problem. Then go back to Step 1.

The branch-and-cut method has been used to solve some of the most difficult problems in integer and combinatorial programming, and a large amount of research has been devoted to improving the performance of the branch-and-cut method and to developing good cutting planes for many important classes of integer and combinatorial problems.