Integer Programming

IE418

Lecture 10

Dr. Ted Ralphs
Reading for This Lecture

• Nemhauser and Wolsey Sections II.1.1-II.1.3, II.1.6
• Wolsey Chapter 8
• Valid Inequalities for Mixed Integer Linear Programs, G. Cornuejols (2006)
**Describing \( \text{conv}(S) \)**

- We now switch to the case of a pure integer program with explicit nonnegativity constraints (the reason will become clear).

\[
 z_{IP} = \max \{ c^\top x \mid x \in S \} = \max \{ c^\top x \mid x \in \text{conv}(S) \},
\]

where

\[
 \mathcal{P} = \{ x \in \mathbb{R}_+^n \mid Ax \leq b \} \quad \text{and} \quad S = \mathcal{P} \cap \mathbb{Z}^n.
\]

- We have just seen that in theory, it would be possible to generate a complete description of \( \text{conv}(S) \).

- So why aren’t IPs easy to solve?
  - The number of inequalities is generally HUGE!
  - The number of facets of the TSP polytope for an instance with 120 nodes is more than \( 10^{100} \) times the number of atoms in the universe.
  - It is physically impossible to write down a description of this polytope.
  - Not only that, but it is very difficult in general to generate these facets (this problem is not polynomially solvable in general).
For Example

• For a TSP of size 15
  – The number of subtour elimination constraints is 16,368.
  – The number of comb inequalities is 1,993,711,339,620.
  – These are only two of the known classes of facets for the TSP.

• For a TSP of size 120
  – The number of subtour elimination constraints is $0.6 \times 10^{36}$!
  – The number of comb inequalities is approximately $2 \times 10^{179}$!
Basic Bounding Methods

• Our discussions of branch and bound has so far focused on the use of three basic bounding methods.
  – LP relaxation
  – Lagrangian relaxation
  – Combinatorial relaxation

• Branch and bound is fundamentally based on the dynamic generation and imposition of valid disjunctions.

• We will now show how disjunctions can also be exploited to generate inequalities valid for \( \text{conv}(S) \).
Cutting Planes

• Recall that the inequality denoted by $(\pi, \pi_0)$ is valid for a polyhedron $P$ if $\pi x \leq \pi_0 \ \forall x \in P$.

• The term cutting plane usually refers to an inequality valid for $\text{conv}(S)$, but which is violated by the solution obtained by solving the (current) LP relaxation.

• Cutting plane methods attempt to improve the bound produced by the LP relaxation by iteratively adding cutting planes to the initial LP relaxation.

• Adding such inequalities to the LP relaxation may improve the bound (this is not a guarantee).
The Separation Problem

- Formally, the problem of generating a cutting plane can be stated as follows.

  **Separation Problem**: Given a polyhedron $\mathcal{P} \subseteq \mathbb{R}^n$ and $x^* \in \mathbb{R}^n$, determine whether $x^* \in \mathcal{P}$ and if not, determine $(\pi, \pi_0)$, a valid inequality for $\mathcal{P}$ such that $\pi x^* > \pi_0$.

- This problem is stated here independent of any solution algorithm.

- However, it is typically used as a subroutine inside an iterative method for improving the LP relaxation.

- In such a case, $x^*$ is the solution to the LP relaxation (of the current formulation, including previously generated cuts).

- We will see later that the difficulty of solving this problem exactly is strongly tied to the difficulty of the optimization problem itself.
Generic Cutting Plane Method

Let $\mathcal{P} = \{ x \in \mathbb{R}^n \mid Ax \leq b \}$ be the initial formulation for

$$\max \{ c^\top x \mid x \in \mathcal{S} \},$$

where $\mathcal{S} = \mathcal{P} \cap \mathbb{Z}^n$ as defined above.

Cutting Plane Method

- Solve the LP relaxation $\max \{ c^\top x \mid x \in \mathcal{P} \}$ to obtain the solution $x^0$.
- Set $\mathcal{P}_0 \leftarrow \mathcal{P}^*$.
- Set $k \leftarrow 0$.
- Iterate
  - Solve the problem of separating $x^k$ from $\mathcal{P}_k$.
  - If $x^k \in \mathcal{P}_k$, then $z_{IP} = c^\top x^k$. STOP.
  - If $x^k \not\in \mathcal{P}_k$, we obtain an inequality $(\pi^k, \pi_0^k)$ valid for $\mathcal{P}$ but for which $\pi^\top x^k > \pi_0^k$.
  - Let $\mathcal{P}_{k+1} = \mathcal{P}_k \cap \{ x \in \mathbb{R}^n \mid \pi^\top x \leq \pi^0 \}$. 
Questions to be Answered

• How do we solve the separation problem?
• Will this algorithm terminate?
• If it does terminate, are we guaranteed to get the optimal solution?
• Note that the algorithm as stated only returns the optimal solution \textit{value}.
• We must obtain an extremal solution to be sure we get an optimal \textit{solution}. 
Methods for Generating Cutting Planes

• Methods for generating cutting planes attempt to solve *separation problem*.

• In most cases, the separation problems that arises cannot be solved exactly, so we either
  – solve the separation problem heuristically, or
  – solve the separation problem exactly, but for a relaxation.

• The *template paradigm* for separation consists of restricting the class of inequalities considered to just those with a specific form.

• This is equivalent, in some sense, to solving the separation problem for a relaxation.

• Separation algorithm can be generally divided into two classes
  – Algorithms that do not assume any specific structure.
  – Algorithms that only work in the presence of specific structure.
Generating Cutting Planes: Two Basic Viewpoints

- There are a number of different points of view from which one can derive the standard methods used to generate cutting planes for general MILPs.

- As we have seen before, there is an *algebraic* point of view and a *geometric* point of view.

- **Algebraic:**
  - Take combinations of the known valid inequalities.
  - Use rounding to produce stronger ones.

- **Geometric:**
  - Use a disjunction (as in branching) to generate several disjoint polyhedra whose union contains $S$.
  - Generate inequalities valid for the convex hull of this union.

- Although these seem like very different approaches, they turn out to be very closely related.
Generating Valid Inequalities: Algebraic Viewpoint

• Consider the feasible region of the LP relaxation \( \mathcal{P} = \{ x \in \mathbb{R}^n_+ \mid Ax \leq b \} \).

• Valid inequalities for \( \mathcal{P} \) can be obtained by taking nonnegative linear combinations of the rows of \((A, b)\).

• Except for one pathological case\(^1\), all valid inequalities for \( \mathcal{P} \) are either equivalent to or dominated by an inequality of the form

\[
\mathbf{u} \mathbf{A} \mathbf{x} \leq \mathbf{u} \mathbf{b}, \mathbf{u} \in \mathbb{R}^m_+.
\]

• To avoid the pathological case, we may assume that \( A \) contains explicit upper bounds on the variables.

\[^1\text{The pathological case occurs when one or more variables have no explicit upper bound and both the primal and dual problems are infeasible.}\]
Generating Valid Inequalities for $\text{conv}(S)$

- All inequalities valid for $\mathcal{P}$ are also valid for $\text{conv}(S)$, but they are not cutting planes.
- We can do better.
- We need the following simple principle: if $a \leq b$ and $a$ is an integer, then $a \leq \lfloor b \rfloor$.
- Believe it or not, this simple fact is all we need to generate all valid inequalities for $\text{conv}(S)$!
Recall again the matching problem.

\[
\begin{align*}
\text{min} & \quad \sum_{e=\{i,j\} \in E} c_e x_e \\
\text{s.t.} & \quad \sum_{\{j\mid \{i,j\} \in E\}} x_{ij} \leq 1, \quad \forall i \in N \\
& \quad x_e \in \{0, 1\}, \quad \forall e = \{i, j\} \in E.
\end{align*}
\]
Generating the Odd Cut Inequalities

- Recall that each odd cutset induces a possible valid inequality.

\[ \sum_{e \in \delta(S)} x_e \geq 1, \ S \subset N, \ |S| \text{ odd.} \]

- Let’s derive these another way.
  - Consider an odd set of nodes \( U \).
  - Sum the constraints \( \sum_{\{j|\{i,j\} \in E\}} x_{ij} \leq 1 \) for \( i \in U \).
  - This results in the inequality \( 2 \sum_{e \in E(U)} x_e + \sum_{e \in \delta(u)} x_e \leq |U| \).
  - Dividing through by 2, we obtain \( \sum_{e \in E(U)} x_e + \frac{1}{2} \sum_{e \in \delta(u)} x_e \leq \frac{1}{2} |U| \).
  - We can drop the second term of the sum to obtain

\[ \sum_{e \in E(U)} x_e \leq \frac{1}{2} |U|. \]

- What’s the last step?
The Chvátal-Gomory Procedure

• Let \( A = (a_1, a_2, \ldots, a_n) \) and \( N = \{1, \ldots, n\} \).

1. Choose a weight vector \( u \in \mathbb{R}^m \).
2. Obtain the valid inequality \( \sum_{j \in N} (ua_j)x \leq ub \).
3. Round the coefficients down to obtain \( \sum_{j \in N} ([ua_j])x \leq ub \). Why can we do this?
4. Finally, round the right hand side down to obtain the valid inequality

\[
\sum_{j \in N} ([ua_j])x \leq [ub]
\]

• This procedure is called the Chvátal-Gomory rounding procedure, or simply the C-G procedure.

• Surprisingly, any inequality valid for \( \text{conv}(S) \) can be produced by a finite number of iterations of this procedure!

• This is not true for the general mixed case.
Assessing the Procedure

• Although it is theoretically possible to generate any valid inequality using the C-G procedure, this is not true in practice.

• The two biggest challenges are numerical errors and slow convergence.

• The inequalities produced may be very weak—we may not even obtain a supporting hyperplane.

• This is because the rounding only “pushes” the inequality until it meets some point in \( \mathbb{Z}^n \), which may or may not even be in \( S \).

• The coefficients of the generated inequality must be relatively prime to ensure the generated hyperplane even includes an integer point!

**Proposition 1.** Let \( S = \{x \in \mathbb{Z}^n \mid \sum_{j \in N} a_j x_j \leq b\} \), where \( a_j \in \mathbb{Z} \) for \( j \in N \), and let \( k = \gcd\{a_1, \ldots, a_n\} \). Then \( \text{conv}(S) = \{x \in \mathbb{R}^n \mid \sum_{j \in N} (a_j/k)x_j \leq \lfloor b/k \rfloor\} \).
Another Viewpoint: Modular Arithmetic

• Let’s consider $T$, the set of solutions to an IP with one equation:

$$ T = \left\{ x \in \mathbb{Z}_+^n \mid \sum_{j=1}^{n} a_j x_j = a_0 \right\} $$

• For each $j$, let $f_j = a_j - \lfloor a_j \rfloor$. Then equivalently

$$ T = \left\{ x \in \mathbb{Z}_+^n \mid \sum_{j=1}^{n} f_j x_j = f_0 + k \text{ for some integer } k \right\} $$

• Since $\sum_{j=1}^{n} f_j x_j \geq 0$ and $f_0 < 1$, then $k \geq 0$ and so

$$ \sum_{j=1}^{n} f_j x_j \geq f_0 $$

is a valid inequality for $S$ called a Gomory cut.
Gomory Cuts from Valid Inequalities

- When solving a general integer program, the set $T$ from the previous slide can be derived from a valid inequality for $P$.

- Choose $\lambda \in \mathbb{R}^m$ so that $(\pi, \pi_0)$ is a valid inequality for $P$, where $\pi = \lambda A$ and $\pi_0 = \lambda b$.

- If we add a slack variable $s$ to convert this inequality into an equality, then the Gomory cut for

\[
T = \left\{(x, s) \in \mathbb{Z}_+^n \times \mathbb{R}_+ \mid \sum_{j=1}^{n} \pi_j x_j + s = \pi_0 \right\}
\]

is

\[
\sum_{j=1}^{n} (\lambda A_j - \lfloor \lambda A_j \rfloor) x_k \geq \lambda b - \lfloor \lambda b \rfloor.
\]

- This is just a C-G inequality with weights $u_i = \lambda_i - \lfloor \lambda_i \rfloor$. 
Gomory Cuts from the Tableau

• Gomory cutting planes can also be derived directly from the tableau while solving an LP relaxation.

• Consider the set

\[ \{(x, s) \in \mathbb{Z}^{n+m}_+ \mid Ax + Is = b\} \]

in which the LP relaxation is put in standard form.

• We assume for now that \( A \) has integral coefficients so that the slack variables also have integer values implicitly.

• The tableau corresponding to basis matrix \( B \) is

\[ B^{-1}Ax + B^{-1}s = B^{-1}b \]

• Each row of this tableau corresponds to a weighted combination of the original constraints.

• The weight vectors are the rows of \( B^{-1} \).
**Gomory Cuts from the Tableau (cont.)**

- A row of the tableau is obtained by combining the equations in the standard representation with weight vector $\lambda = B_j^{-1}$ to obtain

$$\sum_{j=1}^{n}(\lambda A_j)x_j + \sum_{i=1}^{m}\lambda_is_i = \lambda b,$$

where $A_j$ is the $j^{th}$ column of $A$ and $\lambda$ is a row of $B^{-1}$.

- Applying the previous procedure, we can obtain the valid inequality

$$\sum_{j=1}^{n}(\lambda A_j - \lfloor \lambda A_j \rfloor)x_j + \sum_{i=1}^{m}(\lambda_i - \lfloor \lambda_i \rfloor)s_i \geq \lambda b - \lfloor \lambda b \rfloor.$$

- We will show that this Gomory cut is equivalent to the C-G inequality with weights $u_i = \lambda_i - \lfloor \lambda_i \rfloor$. 
Gomory Versus C-G

• To show the Gomory cut is a C-G cut, we first apply the C-G procedure directly to the tableau row, resulting in the inequality

\[
\sum_{j=1}^{n} \lfloor \lambda A_j \rfloor x_j + \sum_{i=1}^{m} \lfloor \lambda_i \rfloor s_i = \lfloor \lambda b \rfloor.
\]

• Note that we can also obtain this inequality by adding the Gomory cut and the original tableau row.

• Now, we substitute out the slack variables using the equation

\[
s = b - Ax.
\]

to obtain

\[
\sum_{j=1}^{n} \left( \lfloor \lambda A_j \rfloor - \sum_{i=1}^{m} \lfloor \lambda_i \rfloor a_{ij} \right) x_j \leq \lfloor \lambda b \rfloor - \sum_{i=1}^{m} \lfloor \lambda_i \rfloor b_i,
\]
Gomory Versus C-G (cont.)

• The final inequality from the previous slide can be re-written as

\[ \sum_{j=1}^{n} \left( \sum_{i=1}^{m} (\lambda_i - \lfloor \lambda_i \rfloor) a_{ij} \right) x_j \leq \sum_{i=1}^{m} (\lambda_i - \lfloor \lambda_i \rfloor) b_i, \]

which is a C-G inequality.

• The substitution of slack variables is more than just a textbook procedure to show the Gomory cut is a C-G cut.

• In practice, the slack variables are substituted out in this fashion in order to derive a cut in terms of the original variables.
Strength of Gomory Cuts from the Tableau

• Consider a row of the tableau in which the value of the basic variable is not an integer.

• Applying the procedure from the last slide, the resulting inequality will only involve nonbasic variables and will be of the form

\[ \sum_{j \in NB} f_j x_j \geq f_0 \]

where \( 0 \leq f_j < 1 \) and \( 0 < f_0 < 1 \).

• The left-hand side of this cut has value zero with respect to the solution to the current LP relaxation.

• We can conclude that the generated inequality will be violated by the current solution to the LP relaxation.
A Finite Cutting Plane Procedure

- Under mild assumptions on the algorithm used to solve the LP, this yields a general algorithm for solving (pure) integer programs.
Example: Gomory Cuts

Consider the polyhedron $\mathcal{P}$ described by the constraints

\begin{align*}
4x_1 + x_2 & \leq 28 \quad (1) \\
x_1 + 4x_2 & \leq 27 \quad (2) \\
x_1 - x_2 & \leq 1 \quad (3) \\
x_1, x_2 & \geq 0 \quad (4)
\end{align*}

Graphically, it can be easily determined that the facet-inducing valid inequalities describing $\text{conv}(S = \mathcal{P} \cap \mathbb{Z}^2)$ are

\begin{align*}
x_1 + 2x_2 & \leq 15 \quad (5) \\
x_1 - x_2 & \leq 1 \quad (6) \\
x_1 & \leq 5 \quad (7) \\
x_2 & \leq 6 \quad (8) \\
x_1 & \geq 0 \quad (9) \\
x_2 & \geq 0 \quad (10)
\end{align*}
Example: Gomory Cuts (cont.)

Figure 1: Convex hull of $S$

(17/3,16/3)
Example: Gomory Cuts (cont.)

Consider the optimal tableau of the LP relaxation of the integer program

\[
\max \{2x_1 + 5x_2 \mid x \in S\},
\]

shown in Table 1.

<table>
<thead>
<tr>
<th>Basic var.</th>
<th>(x_1)</th>
<th>(x_2)</th>
<th>(s_1)</th>
<th>(s_2)</th>
<th>(s_3)</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_2)</td>
<td>0</td>
<td>1</td>
<td>-2/30</td>
<td>8/30</td>
<td>0</td>
<td>16/3</td>
</tr>
<tr>
<td>(s_3)</td>
<td>0</td>
<td>0</td>
<td>-1/3</td>
<td>1/3</td>
<td>1</td>
<td>2/3</td>
</tr>
<tr>
<td>(x_1)</td>
<td>1</td>
<td>0</td>
<td>8/30</td>
<td>-2/30</td>
<td>0</td>
<td>17/3</td>
</tr>
</tbody>
</table>

Table 1: Optimal tableau of the LP relaxation

The associated optimal solution to the LP relaxation is also shown in Figure 1.
Example: Gomory Cuts (cont.)

The Gomory cut from the first row is

$$\frac{28}{30}s_1 + \frac{8}{30}s_2 \geq \frac{1}{3},$$

In terms of $x_1$ and $x_2$, we have

$$4x_1 + 2x_2 \leq 33, \quad \text{(G-C1)}$$
Example: Gomory Cuts (cont.)

The Gomory cut from the second row is

\[
\frac{2}{3}s_1 + \frac{1}{3}s_2 \geq \frac{2}{3},
\]

In terms of \(x_1\) and \(x_2\), we have

\[
3x_1 + 2x_2 \leq 27, \quad \text{(G-C2)}
\]
Example: Gomory Cuts (cont.)

The Gomory cut from the third row is

\[
\frac{8}{30} s_1 + \frac{28}{30} s_2 \geq \frac{2}{3},
\]

In terms of \( x_1 \) and \( x_2 \), we have

\[
x_1 + 2x_2 \leq 16, \quad (G-C3)
\]
Example: Gomory Cuts (cont.)

This picture shows the effect of adding all Gomory cuts in the first round.
C-G Inequalities of Rank 1

• The elementary closure of $\mathcal{P}$ is the intersection of half-spaces defined by inequalities in the set

$$e(\mathcal{P}) = \{(\pi, \pi_0) \mid \pi_j = \lfloor ua_j \rfloor \text{ for } j \in N, \pi_0 = \lfloor ub \rfloor \text{ for some } u \in \mathbb{R}^m \}$$

• The elementary closure is described by all of the nondominated C-G inequalities obtained by combining inequalities in the original formulation.

• Although it is not obvious, one can show that the elementary closure is a polyhedron.

• Optimizing over this polyhedron is difficult ($\mathcal{NP}$-hard) in general.

• By solving a linear program, it can be determined whether a given inequality is rank 1.

**Proposition 2.** If $(\pi, \pi_0) \in e(\mathcal{P})$, then $\pi_0 \geq \lfloor \pi_{0}^{LP} \rfloor$.

• Alternatively, if $\pi \in \mathbb{Z}^n$, the inequality $(\pi, \lfloor \pi_{0}^{LP} \rfloor)$ is rank 1.

• This tells us that the effectiveness of the C-G procedure is strongly tied to the strength of our original formulation.
Example: C-G Rank

- Let's consider the C-G rank of the inequality

\[ x_1 + 2x_2 \leq 15, \]

which is facet-defining for \( \text{conv}(S) \) in our example.

- We have

\[ \max_{x \in S} x_1 + 2x_2 = \frac{49}{3}. \] (11)

- Since \( \lfloor \frac{49}{3} \rfloor = 16 \), we conclude that this is not a rank 1 cut.

- Note that the dual solution to the linear program (11) gives us weights with which to combine the original inequalities to get a C-G cut.

- This is the strongest possible C-G cut of rank 1 with those coefficients.
Generating All Valid Inequalities

• Any valid inequality that can be obtained through iterative application of the C-G procedure (or is dominated by such an inequality) is a C-G inequality.

• For pure integer programs, all valid inequalities are C-G inequalities.

**Theorem 1.** Let \((\pi, \pi_0) \in \mathbb{Z}^{n+1}\) be a valid inequality for \(S = \{x \in \mathbb{Z}_+^n \mid Ax \leq b\} \neq \emptyset\). Then \((\pi, \pi_0)\) is a C-G inequality for \(S\).

• The C-G rank denoted \(r(\pi, \pi_0)\) of an inequality \((\pi, \pi_0)\) valid for \(\mathcal{P}\) is defined recursively as follows.
  – All inequalities valid for the elementary closure \(\mathcal{P}^1 = e(\mathcal{P})\) are rank 1.
  – The polyhedron \(\mathcal{P}^2 = e(\mathcal{P}^1)\) is the rank 2 closure—inequalities valid for it that are not rank 1 are rank 2 inequalities.
  – An inequality is rank \(k\) if it is valid for the rank \(k\) closure \(\mathcal{P}^k = e(\mathcal{P}^{k-1})\) and not for \(\mathcal{P}^{k-1}\).

• Any valid inequality \((\pi, \pi_0)\) for which \(\pi_0 < \lfloor \pi_0^{LP} \rfloor\) has rank at least two.

• The C-G rank of \(\mathcal{P}\) is the maximum rank of any facet-defining inequality of \(\text{conv}(S)\).
Bounding The C-G Rank of a Polyhedron

• For most IPs, the rank of the associated polyhedron is an unbounded function of the dimension.

• **Example:**

  – \( P = \{ x \in \mathbb{R}_+^n \mid x_i + x_j \leq 1 \text{ for } i, j \in \mathbb{N}, i \neq j \} \) and \( S = P^n \cap \mathbb{Z}^n \)
  – \( \text{conv}(S) = \{ x \in \mathbb{R}_+^n \mid \sum_{j \in \mathbb{N}} x_j \leq 1 \} \).
  – \( P \) = \( O(\log n) \).

• For a family of polyhedra with bounded rank, there is a certificate for the validity of any given inequality.

• This leads to a certificate of optimality for the associated optimization problem.

• Hence, it is unlikely that the problem of optimizing over any family of polyhedra with bounded rank is \( \mathcal{NP} \)-hard.

• Conversely, for any \( \mathcal{NP} \)-hard IP, the associated family of polyhedra is likely to have unbounded rank.