Computational Integer Programming

Lecture 11: Cutting Plane Methods

Dr. Ted Ralphs

Describing $\mathrm{conv}(\mathcal{S})$

- We have seen that, in theory, $\mathrm{conv}(\mathcal{S})$ is a polyhedron and has a finite description.
- If we "simply" construct that description, we could turn our MILP into an LP.
- So why aren't IPs easy to solve?
  - The size of the description 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 know classes of facets for the TSP.
- For a TSP of size 120
  - The number of subtour elimination constraints is 0.6×10^{36}!
  - The number of comb inequalities is approximately 2×10^{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 $\text{conv}(\mathcal{S})$.

Cutting Planes

- Recall that the inequality denoted by (π, π_0) is *valid* for a polyhedron $\mathcal{P}$ if $\pi x \leq \pi_0 \; \forall x \in \mathcal{P}$.
- The term *cutting plane* usually refers to an inequality valid for $\mathrm{conv}(\mathcal{S})$, 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).
- Note that when π and π_0 are integer, then π, π_0 is a split disjunction for which $X_2 = \emptyset$.

The Separation Problem

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

  Separation Problem: Given a polyhedron $\mathcal{Q} \subseteq \mathbb{R}^n$ and $x^* \in \mathbb{R}^n$, determine whether $x^* \in \mathcal{Q}$ and if not, determine (π, π_0), a valid inequality for $\mathcal{Q}$ 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

$$\max\{c^\top x \mid x \in \mathcal{S}\}, \quad \text{(MILP)}$$

where $\mathcal{S} = \mathcal{P} \cap \mathbb{Z}^r_+ \times \mathbb{R}^{n-p}_+$, as defined previously.

Cutting Plane Method

$\mathcal{P}_0 \leftarrow \mathcal{P}$
$k \leftarrow 0$
while TRUE **do**
  Solve the LP relaxation $\max\{c^\top x \mid x \in \mathcal{P}_k\}$ to obtain a solution x^k
  Solve the problem of separating x^k from $\text{conv}(\mathcal{S})$
  if $x^k \in \text{conv}(\mathcal{S})$ **then**
    STOP
  else
    Determine an inequality (π^k, π^k_0) valid for $\text{conv}(\mathcal{S})$ but for which $\pi^\top x^k > \pi^k_0$.
  end if
  $\mathcal{P}_{k+1} \leftarrow \mathcal{P}_k \cap \{x \in \mathbb{R}^n \mid (\pi^k)^\top x \leq \pi^k_0\}$.
  $k \leftarrow k + 1$
end while

Questions to be Answered

- How do we solve the separation problem?
- Will this algorithm terminate?
- If it does terminate, are we guaranteed to obtain an optimal 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 $\mathcal{S}$.
  - 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 polyhedron $\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax \leq b\}$.
- 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

$$uAx \leq ub, u \in \mathbb{R}^m_+.$$

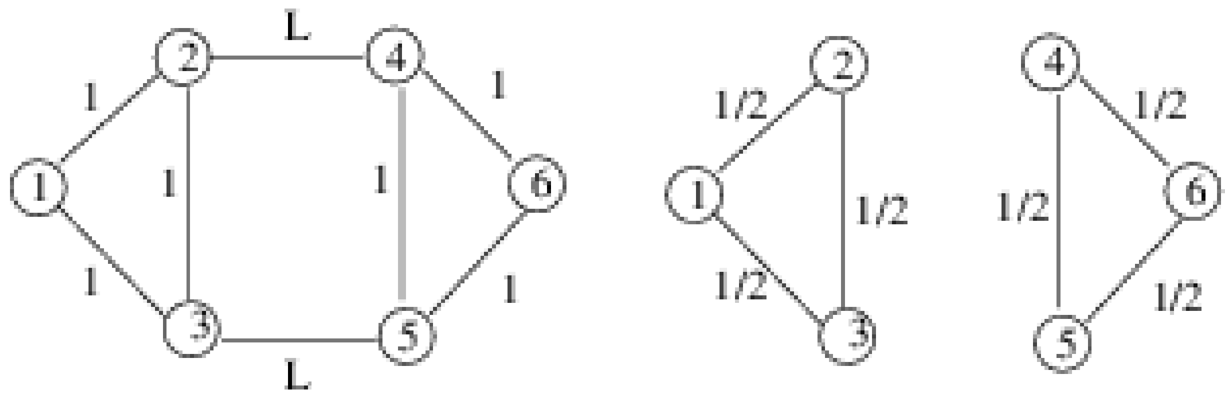
- We are simply taking combinations of inequalities existing in the description, so any such inequalities will be redundant for the LP relaxation.

[1] 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 $\mathrm{conv}(\mathcal{S})$

- All inequalities valid for $\mathcal{P}$ are also valid for $\mathrm{conv}(\mathcal{S})$, 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 $\mathrm{conv}(\mathcal{S})$!

Back to the Matching Problem



Recall again the matching problem.

$$\begin{aligned} \min \quad & \sum_{e=\{i,j\}\in E} c_e x_e \\ s.t. \quad & \sum_{\{j \mid \{i,j\}\in E\}} x_{ij} = 1, \ \forall i \in N \\ & x_e \in \{0,1\}, \qquad \forall e = \{i,j\} \in E. \end{aligned}$$

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 (relaxed) 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?

Chvátal Inequalities

- Suppose we can find a $u \in \mathbb{R}^m_+$ such that $\pi = uA$ is integer and $\pi_0 = ub \notin \mathbb{Z}$.
- In this case, we have $\pi^\top x \in \mathbb{Z}$ for all $x \in \mathcal{S}$, and so $\pi^\top x \leq \lfloor \pi_0 \rfloor$ for all $x \in \mathcal{S}$.
- In other words, $(\pi, \lfloor \pi_0 \rfloor)$ is both a valid inequality and a split disjunction.

Chvátal-Gomory Inequalities

- If we allow the non-negativity constraints to be combined with the constraints of $\mathcal{P}$ (with weight vector v), then integrality of π requires

$$\begin{aligned} uA_I - v_I &\in \mathbb{Z}^p \\ uA_C - v_C &= 0 \end{aligned}$$

so

$$\begin{aligned} &v_i \geq \alpha_i - \lfloor \alpha_i \rfloor && \text{for } 0 \leq i \leq p \\ &v_i = \pi_i \geq 0 && \text{for } p+1 \leq i \leq n \end{aligned}$$

- We then obtain that

$$\sum_{0 \leq i \leq p} \lfloor uA_i \rfloor x_i \leq \lfloor ub \rfloor \qquad \text{(C-G)}$$

is valid for all $u \in \mathbb{R}^m_+$ such that $uA_C \geq 0$.

- This is the *Chvátal-Gomory Inequality*.

The Chvátal-Gomory Procedure

1. Choose a weight vector $u \in \mathbb{R}^m_+$ such that $uA_C \geq 0$.
2. Obtain the valid inequality $\sum_{0 \leq i \leq p} (uA_i) x_i \leq ub$.
3. Round the coefficients down to obtain $\sum_{0 \leq i \leq p} (\lfloor uA_i \rfloor) x_i \leq ub$. Why can we do this?
4. Finally, round the right hand side down to obtain the valid inequality

$$\sum_{0 \leq i \leq p} (\lfloor uA_i \rfloor) x_i \leq \lfloor ub \rfloor$$

- This procedure is called the *Chvátal-Gomory* rounding procedure, or simply the *C-G procedure*.
- Surprisingly, for pure ILPs ($p = n$), any inequality valid for $\text{conv}(\mathcal{S})$ 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 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 $\mathcal{S}$.
- The coefficients of the generated inequality must be relatively prime to ensure the generated hyperplane even includes an integer point!

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

Gomory Inequalities

- Let's consider T, the set of solutions to a pure ILP with one equation:

$$T = \left\{ x \in \mathbb{Z}^n_+ \;\middle|\; \sum_{j=1}^{n} a_j x_j = a_0 \right\}$$

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

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

- 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 $\mathcal{S}$ called a *Gomory inequality*.

- Note that this has been derived as a split disjunction with $X_2 = \emptyset$.

Gomory Cuts from the Tableau

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

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

in which the LP relaxation of an ILP 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

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

where A_j is the j^{th} column of A and λ 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

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

- 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) ILPs.

Example: Gomory Cuts

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

$$4x_1 + x_2 \leq 28 \tag{1}$$

$$x_1 + 4x_2 \leq 27 \tag{2}$$

$$x_1 - x_2 \leq 1 \tag{3}$$

$$x_1, x_2 \geq 0 \tag{4}$$

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

$$x_1 + 2x_2 \leq 15 \tag{5}$$

$$x_1 - x_2 \leq 1 \tag{6}$$

$$x_1 \leq 5 \tag{7}$$

$$x_2 \leq 6 \tag{8}$$

$$x_1 \geq 0 \tag{9}$$

$$x_2 \geq 0 \tag{10}$$

Example: Gomory Cuts (cont.)

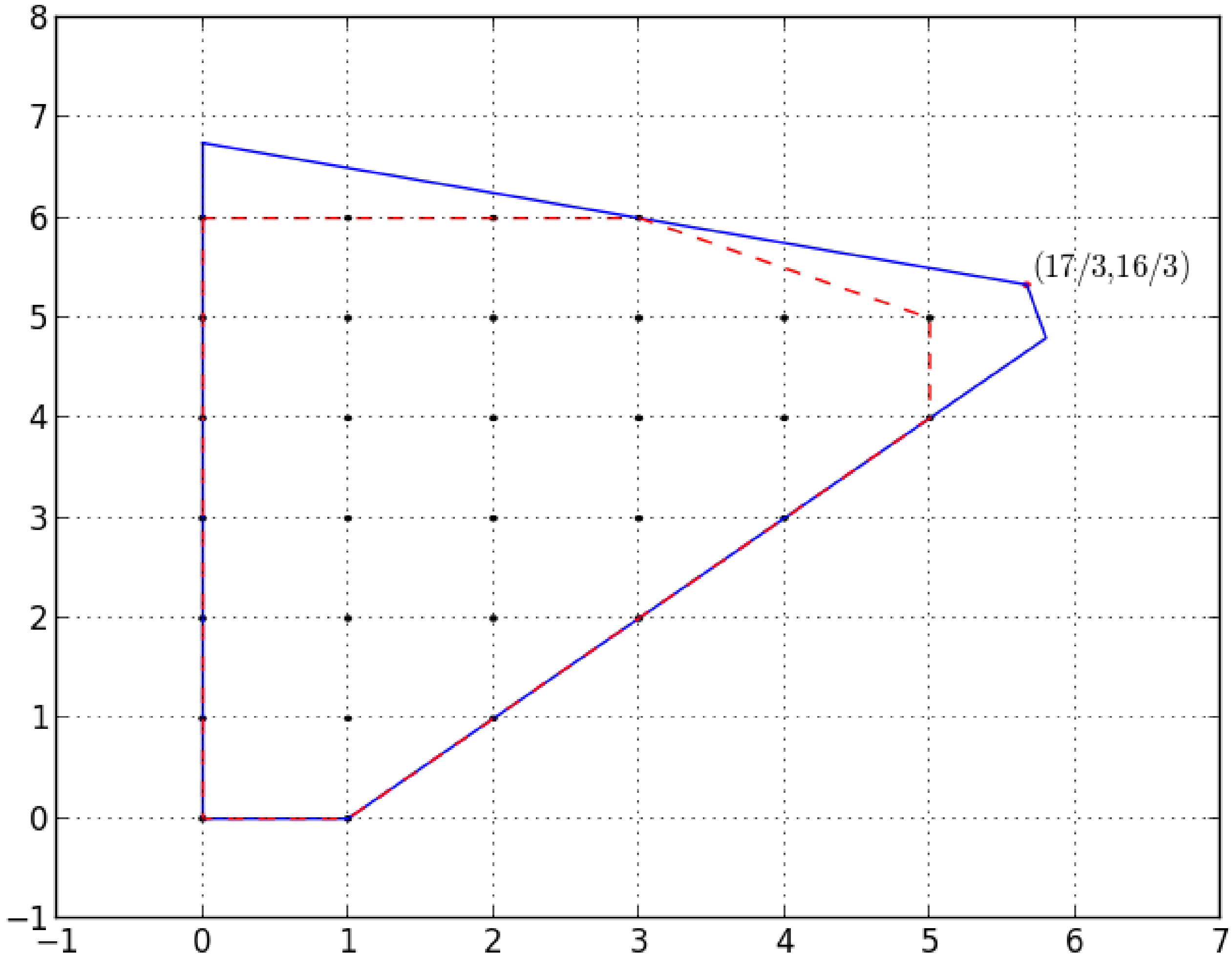


Figure 1: Convex hull of $\mathcal{S}$

Example: Gomory Cuts (cont.)

Consider the optimal tableau of the LP relaxation of the ILP

$$\max\{2x_1 + 5x_2 \mid x \in \mathcal{S}\},$$

shown in Table 2.

Basic var.	x_1	x_2	s_1	s_2	s_3	RHS
x_2	0	1	-2/30	8/30	0	16/3
s_3	0	0	-1/3	1/3	1	2/3
x_1	1	0	8/30	-2/30	0	17/3

Table 1: Optimal tableau of the LP relaxation

The associated optimal solution to the LP relaxation is also shown in Figure 2.

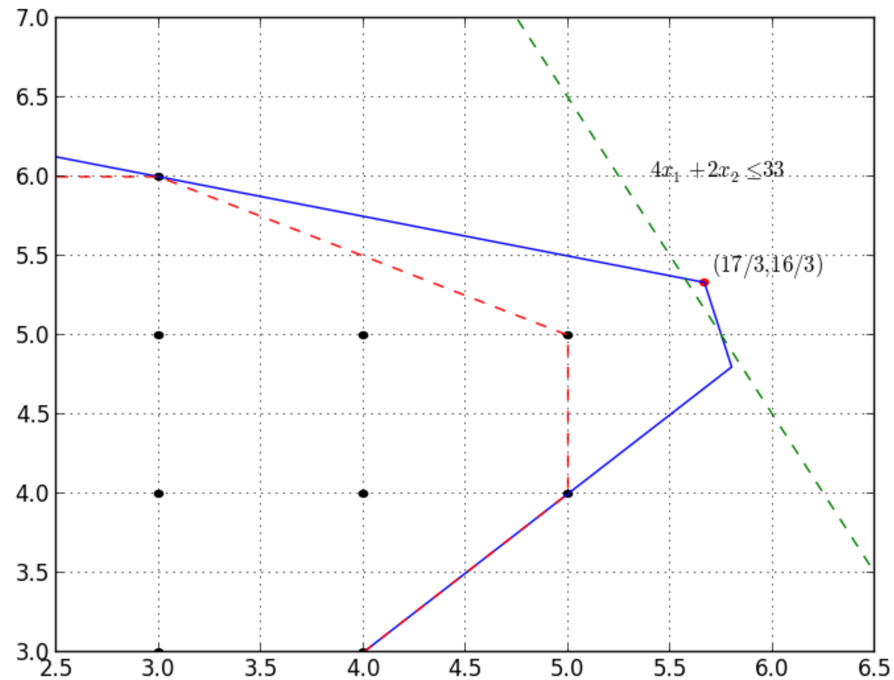
Example: Gomory Cuts (cont.)

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

$$4x_1 + 2x_2 \leq 33, \qquad \text{(G-C1)}$$



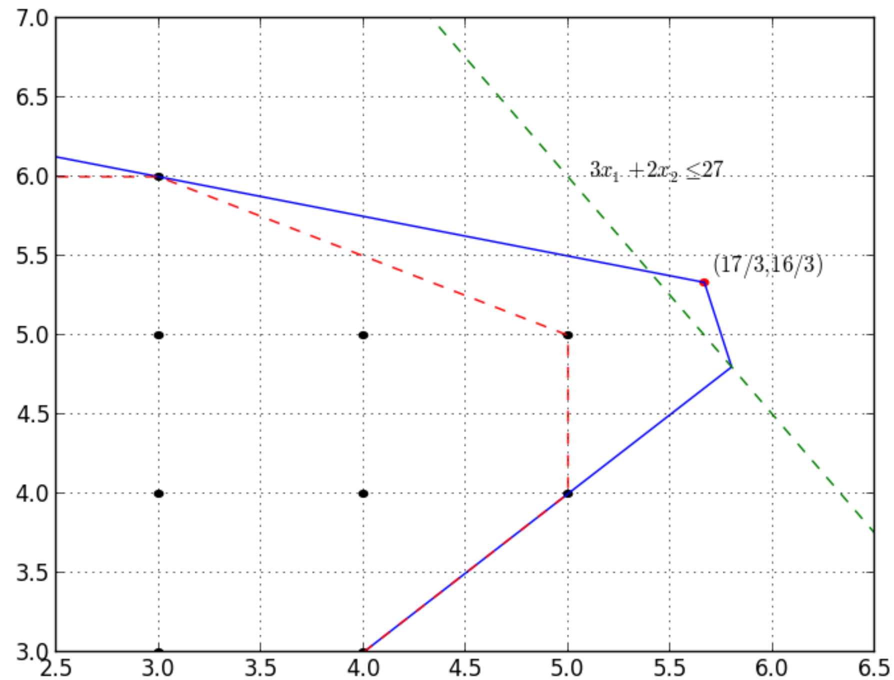
Example: Gomory Cuts (cont.)

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

$$3x_1 + 2x_2 \leq 27, \qquad \text{(G-C2)}$$



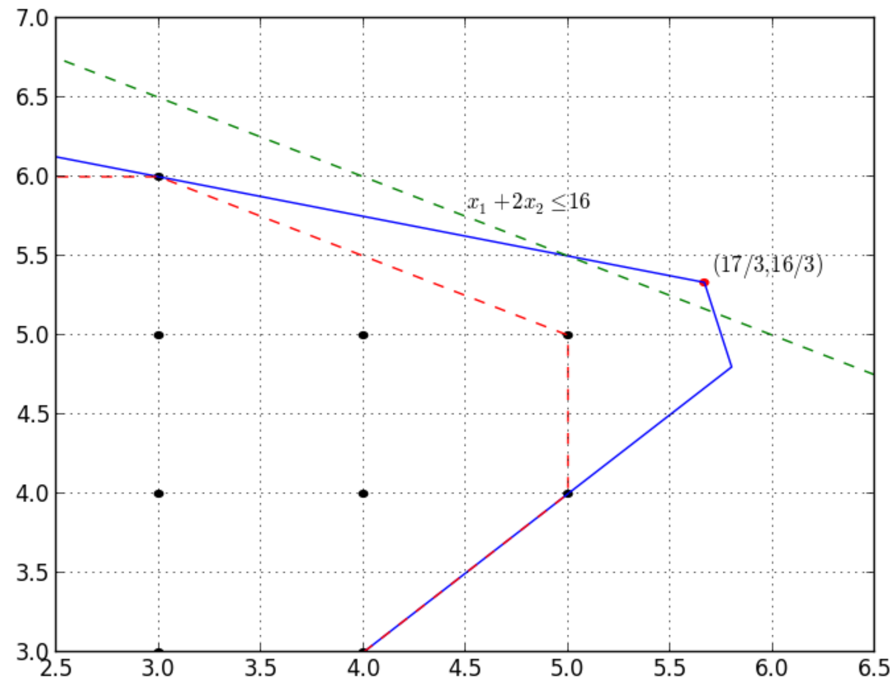
Example: Gomory Cuts (cont.)

The Gomory cut from the third row is

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

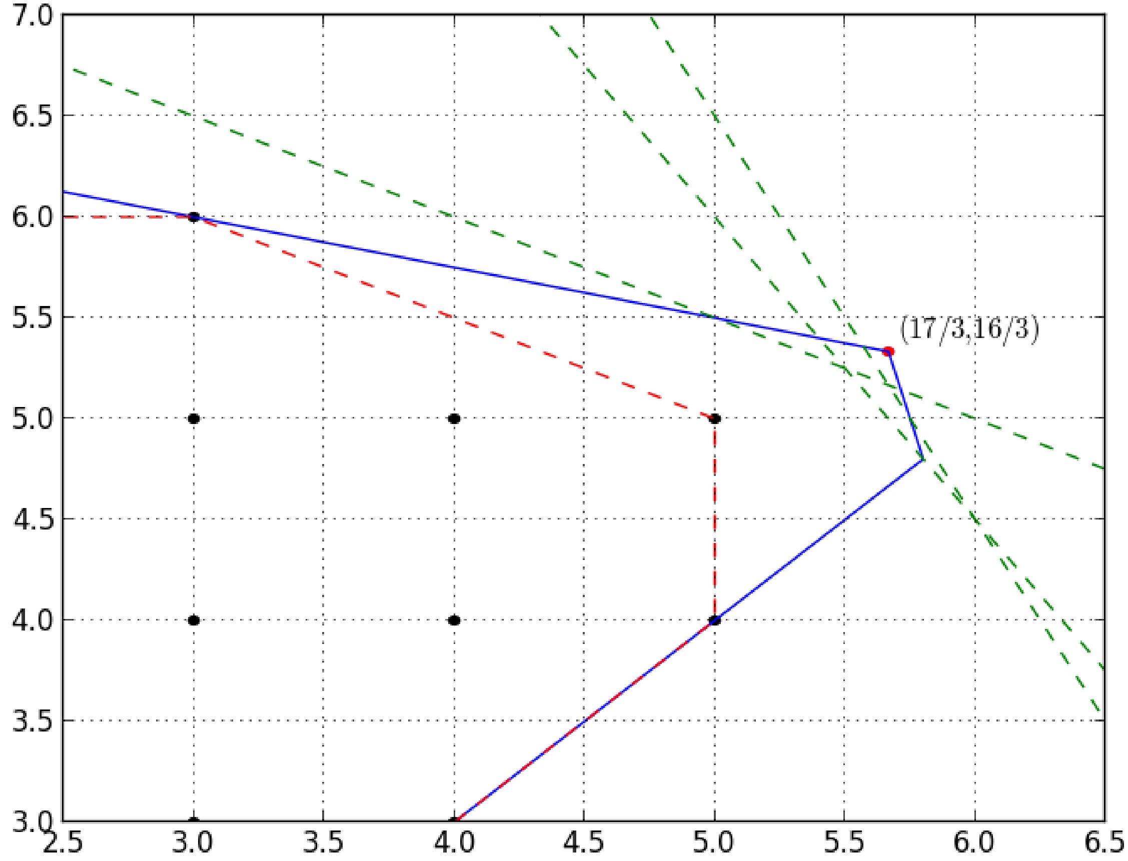
In terms of x_1 and x_2, we have

$$x_1 + 2x_2 \leq 16, \qquad \text{(G-C3)}$$



Example: Gomory Cuts (cont.)

This picture shows the effect of adding all Gomory cuts in the first round.



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 ILPs, all valid inequalities are C-G inequalities.

Theorem 1. *Let $(\pi, \pi_0) \in \mathbb{Z}^{n+1}$ be a valid inequality for $\mathcal{S} = \{x \in \mathbb{Z}^n_+ \mid Ax \leq b\} \neq \emptyset$. Then (π, π_0) is a C-G inequality for $\mathcal{S}$.*

- The C-G rank denoted $r(\pi, \pi_0)$ of an inequality (π, π_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 $\mathcal{P}^{k-1}$.
- The *C-G rank* of $\mathcal{P}$ is the maximum rank of any facet-defining inequality of $\text{conv}(\mathcal{S})$.

Valid Inequalities from Disjunctions

- Valid inequalities for $\text{conv}(\mathcal{S})$ can also be generated based on disjunctions.
- Let $\mathcal{P}_i = \{x \in \mathbb{R}^n_+ \mid A^i x \leq b^i\}$ for $i = 1, \ldots, k$ be such that $\mathcal{S} \subseteq \cup_{i=1}^k \mathcal{P}_i$.
- Then inequalities valid for $\cup_{i=1}^k \mathcal{P}_i$ are also valid for $\text{conv}(\mathcal{S})$.
- The following procedure shows how to generate such inequalities.

Proposition 2. *If (π^1, π_0^1) is valid for $\mathcal{S}_1 \subseteq \mathbb{R}^n_+$ and (π^2, π_0^2) is valid for $\mathcal{S}_2 \subseteq \mathbb{R}^n_+$, then*

$$\sum_{j=1}^n \min(\pi_j^1, \pi_j^2) x_j \leq \max(\pi_0^1, \pi_0^2) \tag{11}$$

for $x \in \mathcal{S}_1 \cup \mathcal{S}_2$.

- In fact, all valid inequalities for the union of two polyhedra can be obtained in this way.

Proposition 3. *If $\mathcal{P}^i = \{x \in \mathbb{R}^n_+ \mid A^i x \leq b^i\}$ for $i = 1, 2$ are nonempty polyhedra, then (π, π_0) is a valid inequality for $\text{conv}(\mathcal{P}^1 \cup \mathcal{P}^2)$ if and only if there exist $u^1, u^2 \in \mathbb{R}^m$ such $\pi \leq u^i A^i$ and $\pi_0 \geq u^i b^i$ for $i = 1, 2$.*

Gomory Mixed Integer Inequalities

- Let's consider again the set of solutions T to an IP with one equation.
- This time, we write T equivalently as

$$T = \left\{ x \in \mathbb{Z}^n_+ \;\middle|\; \sum_{j:f_j \le f_0} f_j x_j + \sum_{j:f_j > f_0} (f_j - 1) x_j = f_0 + k \text{ for some integer k} \right\}$$

- Since $k \le -1$ or $k \ge 0$, we have the disjunction

$$\sum_{j:f_j \le f_0} \frac{f_j}{f_0} x_j - \sum_{j:f_j > f_0} \frac{(1 - f_j)}{f_0} x_j \ge 1$$

OR

$$- \sum_{j:f_j \le f_0} \frac{f_j}{(1 - f_0)} x_j + \sum_{j:f_j > f_0} \frac{(1 - f_j)}{(1 - f_0)} x_j \ge 1$$

The Gomory Mixed Integer Cut

- Applying Proposition 2, we get

$$\sum_{j:f_j \leq f_0} \frac{f_j}{f_0} x_j + \sum_{j:f_j > f_0} \frac{(1-f_j)}{(1-f_0)} x_j \geq 1$$

- This is called a *Gomory mixed integer* (GMI) cut.
- GMI cuts dominate the associated Gomory cut and can also be obtained easily from the tableau.
- In the case of the mixed integer set

$$T = \left\{ x \in \mathbb{Z}^p_+ \times \mathbb{R}^{n-p}_+ \;\middle|\; \sum_{j=1}^{n} a_j x_j = a_0 \right\},$$

the GMI cut is

$$\sum_{\substack{0 \leq j \leq p \\ f_j \leq f_0}} \frac{f_j}{f_0} x_j + \sum_{\substack{0 \leq j \leq p \\ f_j > f_0}} \frac{(1-f_j)}{(1-f_0)} x_j + \sum_{\substack{p+1 \leq j \leq n \\ a_j > 0}} \frac{a_j}{f_0} x_j - \sum_{\substack{p+1 \leq j \leq n \\ a_j < 0}} \frac{a_j}{(1-f_0)} x_j \geq 1$$

Gomory Mixed Integer Cuts from the Tableau

- Let's consider how to generate mixed integer Gomory cuts from the tableau when solving an MILP of the form

$$Q = \{x \in \mathbb{Z}^p_+ \times \mathbb{R}^{n-p}_+ \mid Ax \leq b\}.$$

- We first introduce a slack variable for each inequality in the formulation.
- Solving the LP relaxation, we look for a row in the tableau in which an integer variable is basic and has a fractional variable.
- We apply the GMI procedure to produce a cut.
- Finally, we substitute out the slack variables in order to express the cut in terms of the original variables only.

Example: GMI Cuts versus Gomory Cuts

Recall our example from last time.

$$\max \quad 2x_1 + 5x_2 \tag{12}$$
$$\text{s.t.} \quad 4x_1 + x_2 \leq 28 \tag{13}$$
$$x_1 + 4x_2 \leq 27 \tag{14}$$
$$x_1 - x_2 \leq 1 \tag{15}$$
$$x_1, x_2 \geq 0 \tag{16}$$

The optimal tableau for the LP relaxation is:

Basic var.	x_1	x_2	s_1	s_2	s_3	RHS
x_2	0	1	-2/30	8/30	0	16/3
s_3	0	0	-1/3	1/3	1	2/3
x_1	1	0	8/30	-2/30	0	17/3

Table 2: Optimal tableau of the LP relaxation

The associated optimal solution to the LP relaxation is also shown in Figure 2.

Example: Gomory Cuts (cont.)

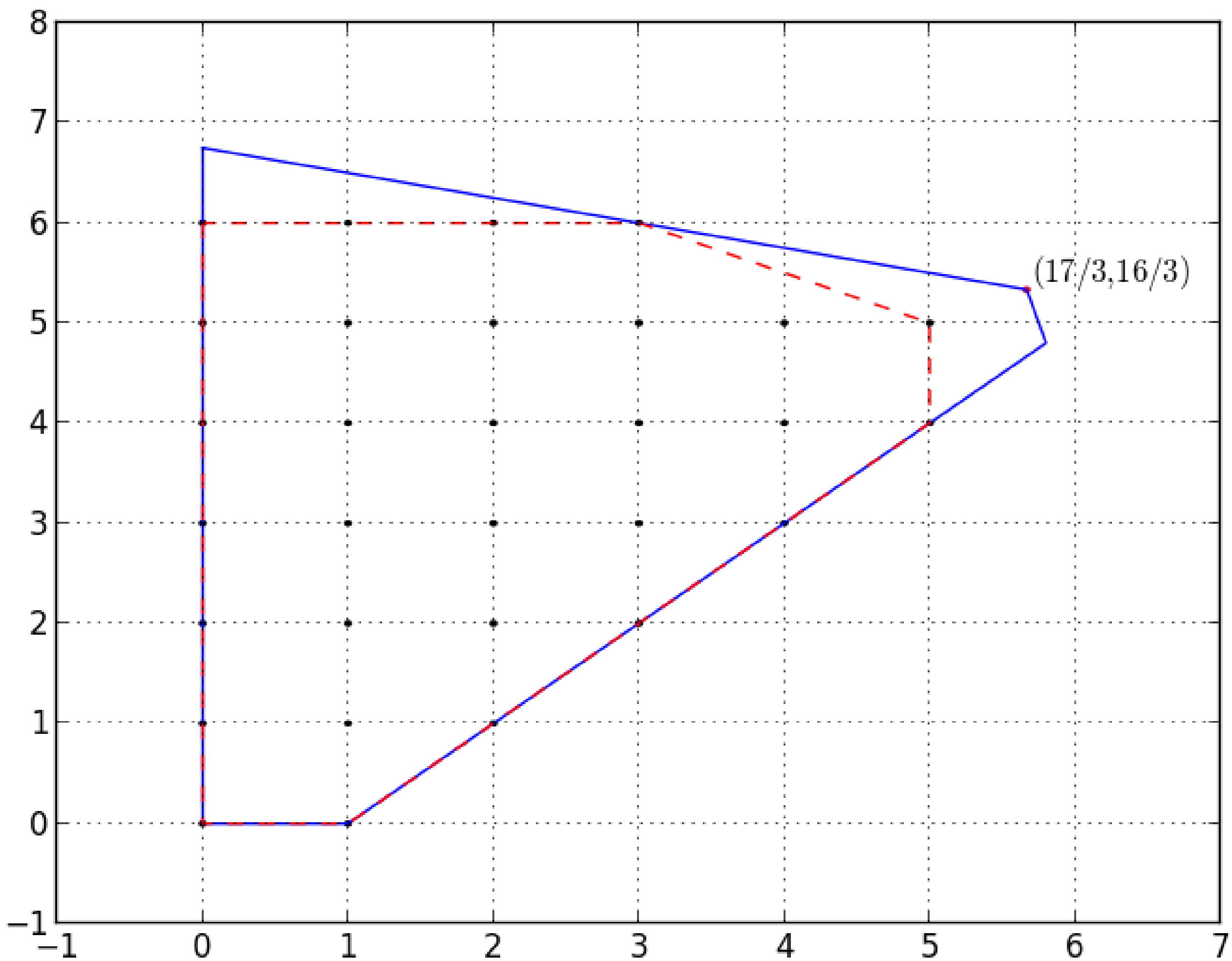


Figure 2: Convex hull of $\mathcal{S}$

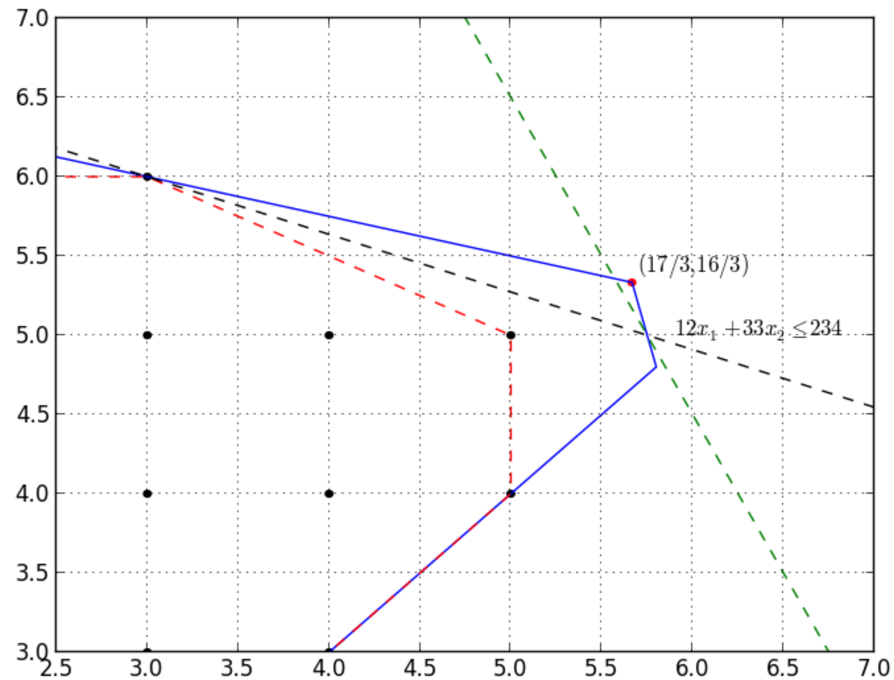
Example: GMI Cuts versus Gomory Cuts (cont.)

The GMI cut from the first row is

$$\frac{1}{10}s_1 + \frac{8}{10}s_2 \geq 1,$$

In terms of x_1 and x_2, we have

$$12x_1 + 33x_2 \leq 234, \tag{GMI-C1}$$



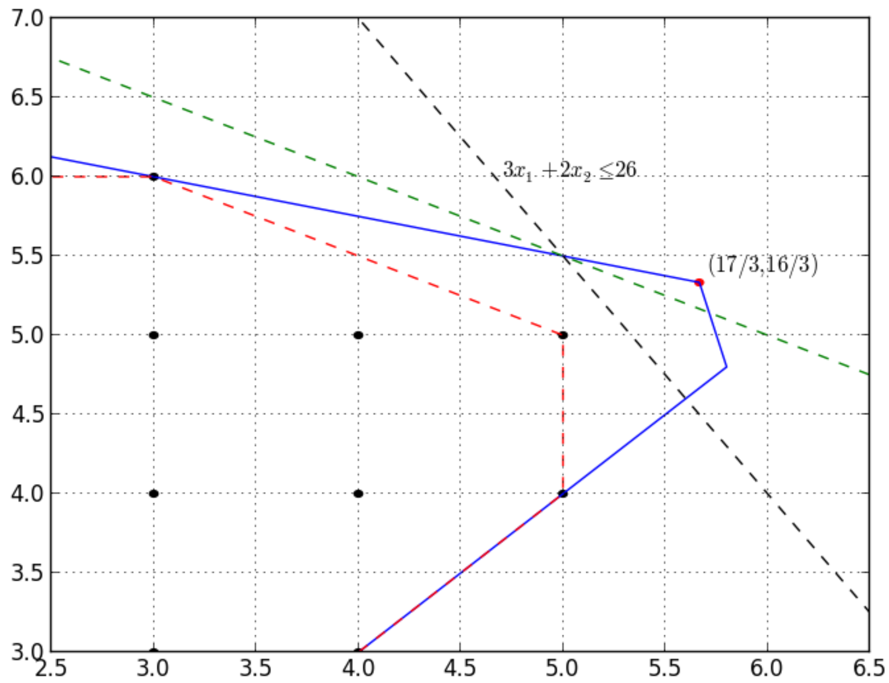
Example: GMI Cuts versus Gomory Cuts (cont.)

The GMI cut from the third row is

$$\frac{4}{10}s_1 + \frac{2}{10}s_2 \geq 1,$$

In terms of x_1 and x_2, we have

$$3x_1 + 2x_2 \leq 26, \qquad \text{(GMI-C3)}$$



Geometric Interpretation of GMI Cuts

- To understand the geometric interpretation of GMI cuts, we consider a relaxation of (**??**) associated with a basis of the LP relaxation.
- We simply relax the non-negativity constraints on the basic variables to obtain

$$T = \left\{(x, s) \in \mathbb{Z}^{n+m} \mid Ax + Is = b, x_N \geq 0, s_N \geq 0\right\},$$

  where x_N and s_N are the non-basic variables associated with basis B.
- This is equivalent to relaxing the non-binding constraints.
- The convex hull of T is the so-called *corner polyhedron* associated with the basis B.

Example: Corner Polyhedron

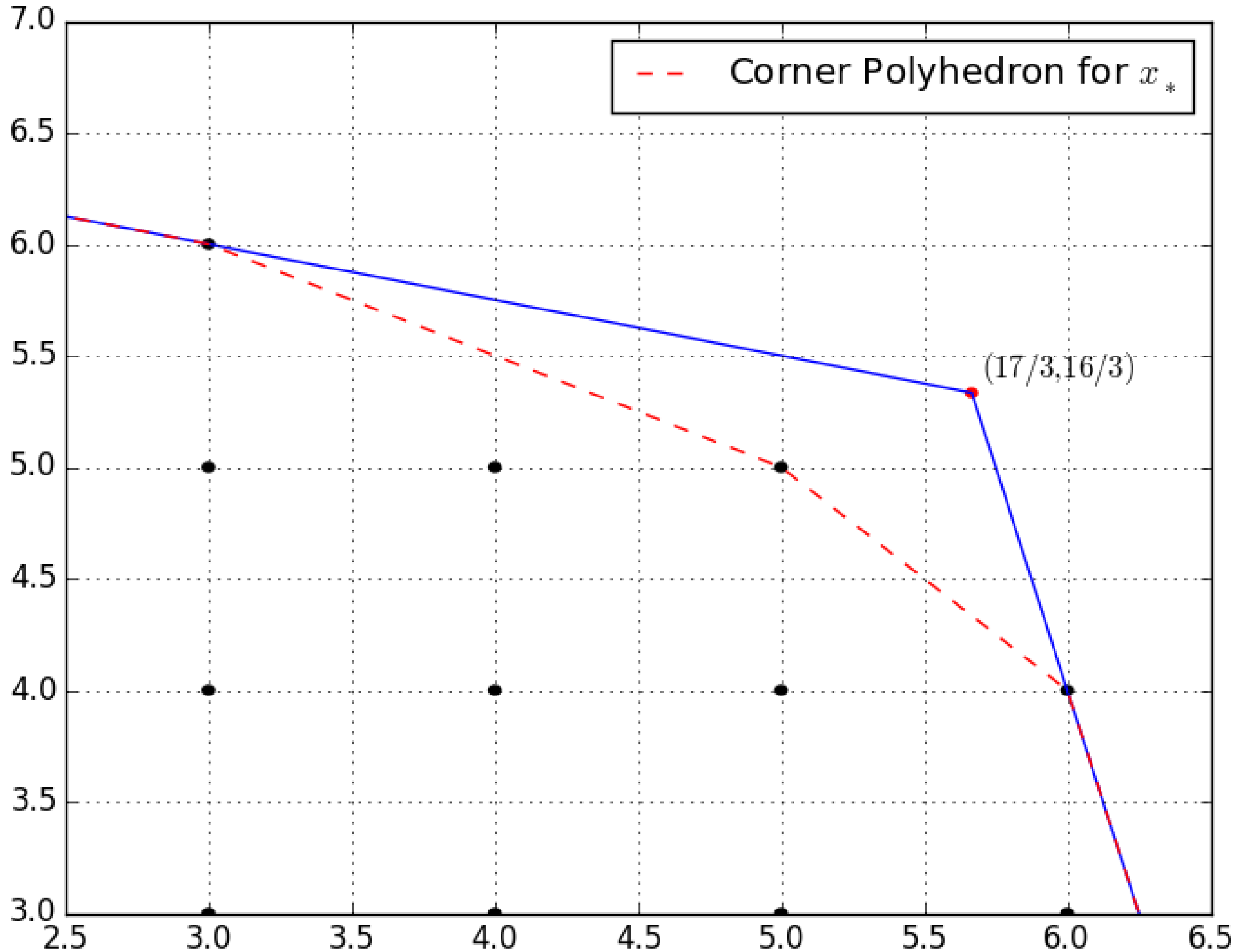


Figure 3: The corner polyhedron associated with the optimal basis of the LP relaxation of the earlier example.

GMI Cuts in Practice

Here is an example of the slow convergence sometimes seen in practice.

$$
\begin{aligned}
\min \quad & 20x_1 + 15x_2 \\
& -2x_1 - 3x_2 \leq -5 \\
& -4x_1 - 2x_2 \leq -15 \\
& -3x_1 - 4x_2 \leq 20 \\
& 0 \leq x_1 \leq 9 \\
& 0 \leq x_2 \leq 6 \\
& x_1, x_2 \in \mathbb{Z}
\end{aligned}
$$

We will solve this using the naive implementation in `CuPPy`.

The Polyhedra in Example

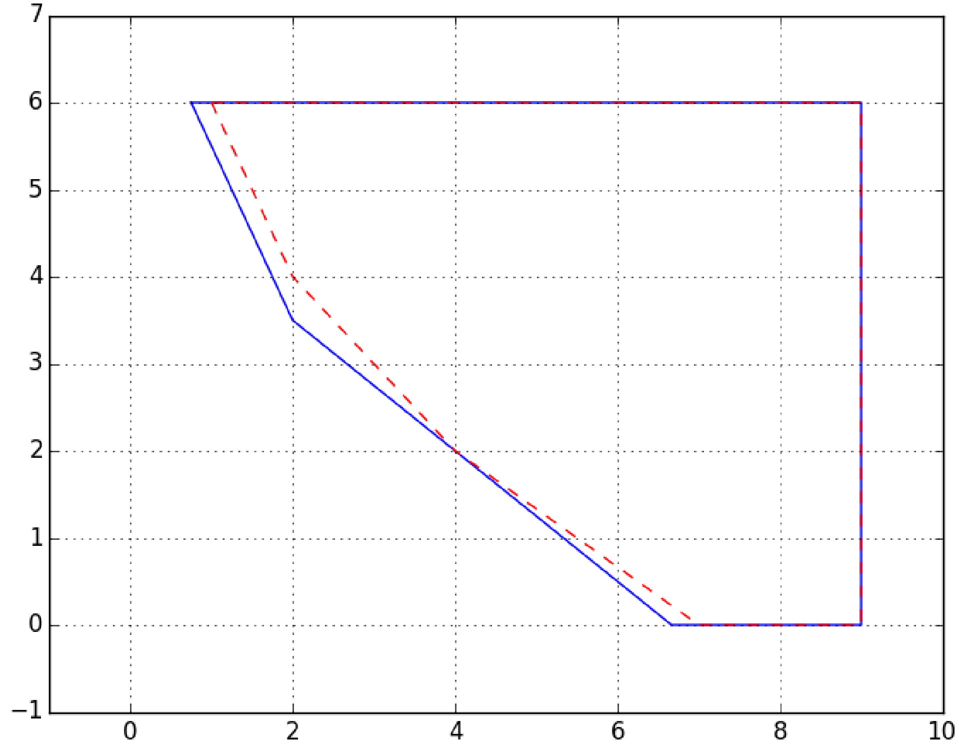


Figure 4: Feasible region of Example MILP

First Iteration

- The solution to the LP relaxation is $(2, 3.5)$.
- The tableau row in which x_2 is basic is

$$x_2 + 0.3s_2 - 0.4s_3$$

- Note that for purposes of illustration, we are explicitly included the bound constraints in the tableau.
- The GMI is

$$0.6s_2 + 0.8s_3 \geq 1$$

- In terms of the original variables, this is

$$-4.8x_1 - 4.4x_2 \leq -26$$

Second Iteration

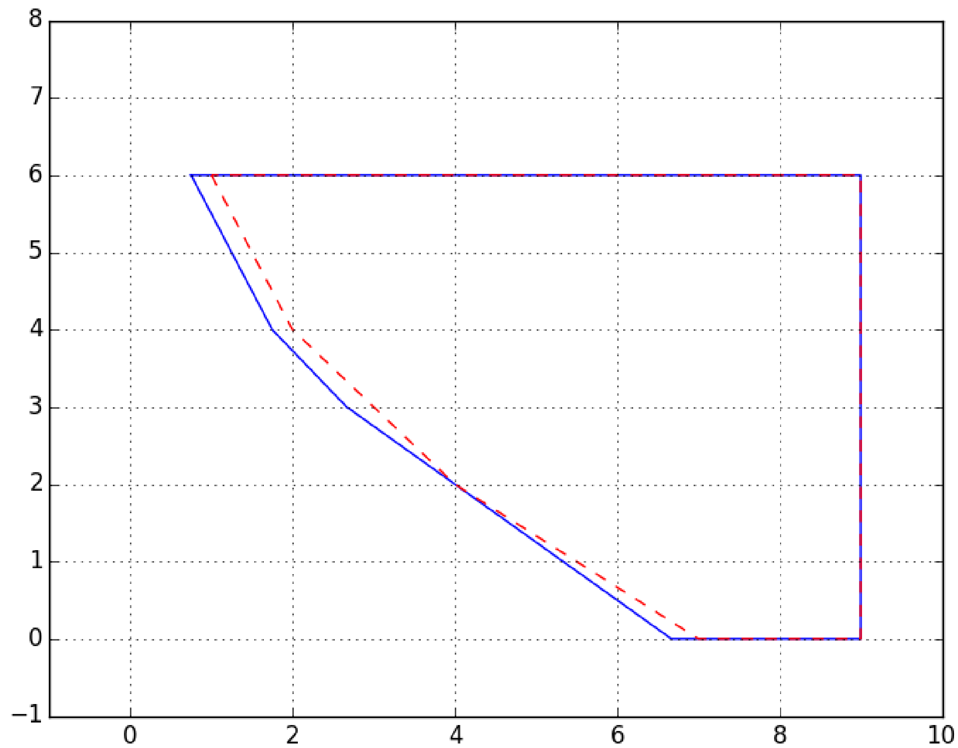


Figure 5: Feasible region of Example MILP after adding cut

The solution in the second iteration is $(1.75, 4)$ and the cut is $-10.4x_1 - 5.8667x_2 \leq -42.6667$.

Third Iteration

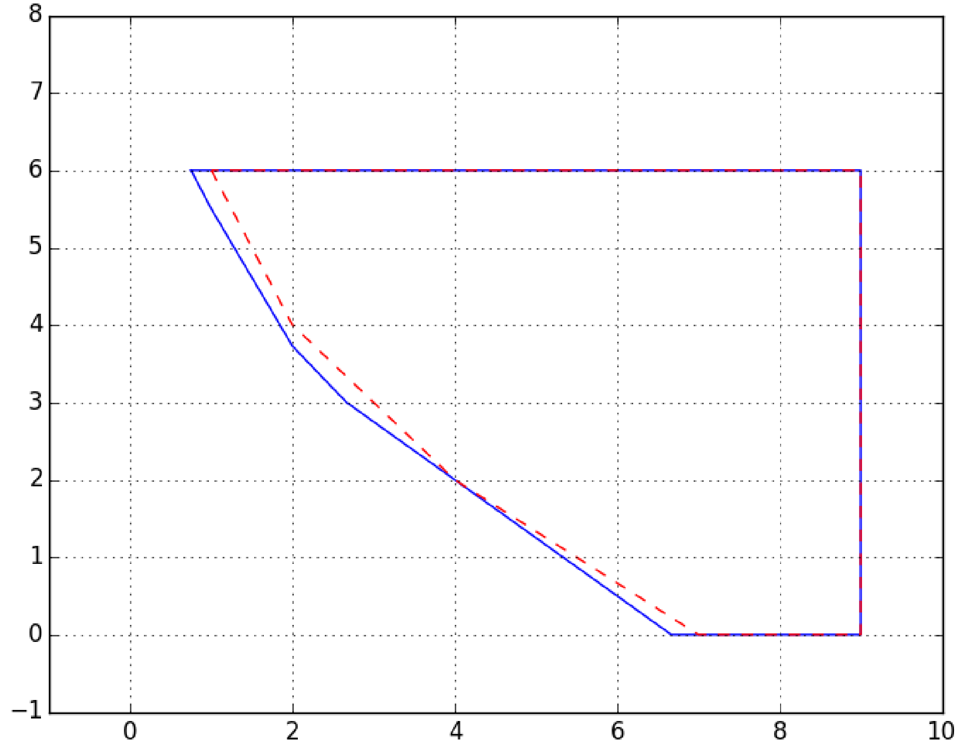


Figure 6: Feasible region of Example MILP after two cuts

The solution in the third iteration is $(2, 3.7273)$ and the cut is $-14.3x_1 - 11.7333x_2 \leq -73.3333$.

Further Iterations

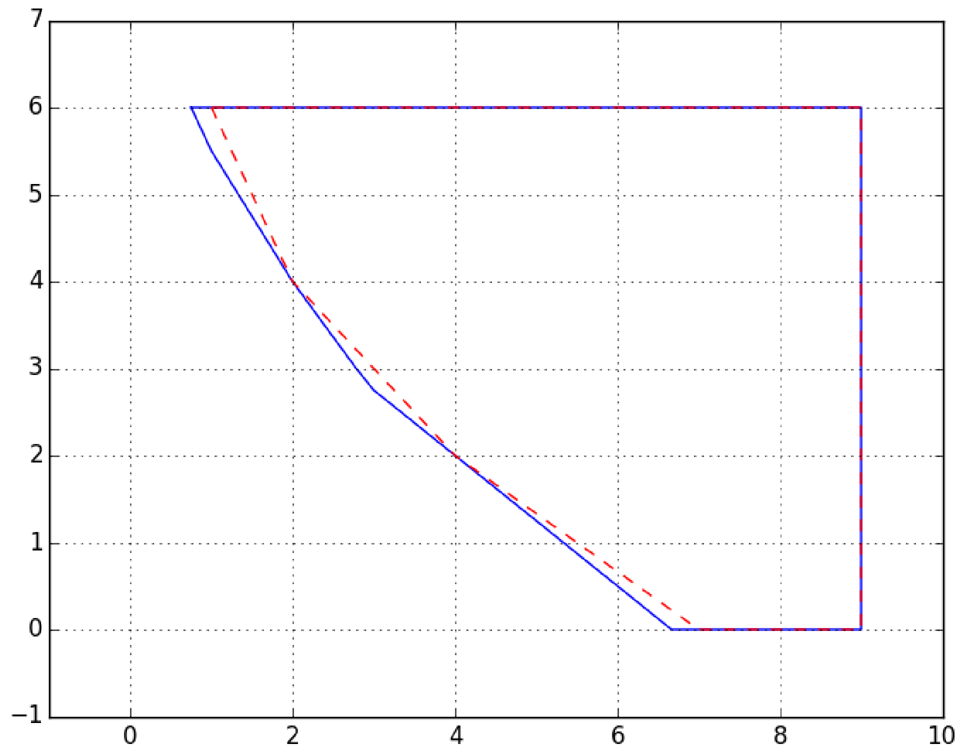


Figure 7: Feasible region of Example MILP after 100 cuts

Further Iterations

- Note the slow convergence rate.
- Not much progress is being made with each cut.
- After 100 iteration, the solution is $(1.9979, 4)$, which may be "close enough," but would not be considered optimal by most solvers.
- It is surprising that such a small MILP would have such a high rank.
- This is at least partly due to numerical errors and the fact that our implementation is naive.
- We will delve further into these topics later in the course.

Split Inequalities

- Let (α, β) be a split disjunction and define

$$\begin{aligned} \mathcal{P}_1 &= \mathcal{P} \cap \{x \in \mathbb{R}^n \mid \alpha^\top x \leq \beta\} \\ \mathcal{P}_2 &= \mathcal{P} \cap \{x \in \mathbb{R}^n \mid \alpha^\top x \geq \beta + 1\} \end{aligned}$$

- Any inequality valid for $\mathrm{conv}(\mathcal{P}_1 \cup \mathcal{P}_2)$ is valid for $\mathcal{S}$ and is called a *split inequality*.

Separation Problem for Split Inequalities

- The LP (**??**) can be generalized straightforwardly to produce the most violated split cut.

$$
\begin{aligned}
\max \quad & \pi\hat{x} - \pi^0 \\
\text{s.t.} \quad & \pi \le uA + u^0\alpha, \\
& \pi \le vA - v^0\alpha, \\
& \pi^0 \ge ub + u_0\beta, \\
& \pi^0 \ge vb - v_0(\beta+1), \\
& \sum_{i=1}^{m} u_i + u_0 + \sum_{i=1}^{m} v_i + v_0 = 1 \\
& u, u_0, v, v_0 \ge 0 \\
& \alpha \in \mathbb{Z}^n \\
& \beta \in \mathbb{Z}
\end{aligned}
\qquad \text{(SCGLP)}
$$

- The separation problem is a mixed integer nonlinear optimization problem, however, and is not easy to solve.

Exercise: Experiment with Cutting Planes

- Download http://miplib.zib.de/download/miplib2010-1.1.2-benchmark zip
- Try solving some instances with and without cutting planes using Cbc.
- Experient with particular classes of cuts.
- A link to instructions for how to use Cbc on the command line is provided on the course page.