# Integer Programming
# ISE 418

## Lecture 13

Dr. Ted Ralphs

# Reading for This Lecture

- Nemhauser and Wolsey Sections II.1.1-II.1.3, II.1.6
- Wolsey Chapter 8
- CCZ Chapters 5 and 6
- "Valid Inequalities for Mixed Integer Linear Programs," G. Cornuejols.
- "Generating Disjunctive Cuts for Mixed Integer Programs," M. Perregaard.

# Valid Inequalities from Disjunctions

- Valid inequalities for $\mathrm{conv}(\mathcal{S})$ can also be generated from valid disjunctions.
- Let $X_i = \{x \in \mathbb{R}^n_+ \mid A^i x \leq b^i\}$ for $i = 1, \ldots, k$ constitute a disjunction valid for $\mathcal{S}$.
- Then inequalities valid for $\cup_{i=1}^k (\mathcal{P} \cap X_i$ are also valid for $\mathrm{conv}(\mathcal{S})$.

# The Union of Polyhedra

- The convex hull of the union of polyhedra is not necessarily a polyhedron.
- Under mild conditions, we can characterize it, however.
- Consider a finite collection of polyhedra $\mathcal{P}_i = \{x \in \mathbb{R}^n \mid A^i x \leq b^i$ for $1 \leq i \leq k$.
- Let $Y$ be the polyhedron described by the following constraints:

$$\begin{aligned} A^i x^i &\leq b^i y_i \ \forall i = 1, \ldots, k \\ \sum_{i=1}^{k} x^i &= x \\ \sum_{i=1}^{k} y^i &= 1 \\ y &\geq 0 \end{aligned}$$

- Furthermore, for polyhedron $\mathcal{P}_i$, let $C_i = \{x \in \mathbb{R}^n \mid A^i x \leq 0\}$ and let $\mathcal{P}_i = Q_i + C_i$ where $Q_i$ is a polytope.

# The Convex Hull of the Union of Polyhedra

- Under the assumptions on the previous slide, we have the following result.

  **Proposition 1.** *If either* $\cup_{i=1}^k \mathcal{P}_i = \emptyset$ *or* $C_j \subseteq \mathrm{cone}(\cup_{i:\mathcal{P}_i \neq \emptyset} C_i$ *for all* $j$ *such that* $\mathcal{P}_j = \emptyset$*, then the following sets are identical:*

  - $\overline{\mathrm{conv}}(\cup_{i=1}^k \mathcal{P}_i\}$
  - $\mathrm{conv}(\cup_{i=1}^k Q_i) + cone(\cup_{i=1}^k C_i)$
  - $proj_x Y$.

- Note that the assumptions of the proposition are necessary, but are automatically satisfied if

  - $C^i = \{0\}$ whenever $\mathcal{P}^i = \emptyset$, or
  - all the polyhedra have the same recession cone.

# The Convex Hull of the Union of Polyhedra (cont.)

- Note also that if all the polyhedra have the same recession cones, then $\overline{\text{conv}}(\cup_{i=1}^k \mathcal{P}_i) = \text{conv}(\cup_{i=1}^k \mathcal{P}_i)$ and $\cup_{i=1}^k \mathcal{P}_i$ is the projection of

$$
\begin{aligned}
A^i x^i &\leq b^i y_i \ \ \forall i = 1, \ldots, k \\
\sum_{i=1}^k x^i &= x \\
\sum_{i=1}^k y^i &= 1 \\
y &\in \{0, 1\}
\end{aligned}
$$

- This is the case when the polyhedra only differ in their right-hand sides, as is the case when branching on variables.

# Valid Inequalities from Disjunctions

Another viewpoint for constructing valid inequalities based on disjunctions comes from the following result:

**Proposition 2.** *If* $(\pi^1, \pi_0^1)$ *is valid for* $\mathcal{S}_1 \subseteq \mathbb{R}_+^n$ *and* $(\pi^2, \pi_0^2)$ *is valid for* $\mathcal{S}_2 \subseteq \mathbb{R}_+^n$*, then*

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

*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* $(\pi, \pi_0)$ *is a valid inequality for* $\mathrm{conv}(\mathcal{P}^1 \cup \mathcal{P}^2)$ *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* $i = 1, 2$*.*

# Simple Disjunctive Inequalities

- We want to develop a procedure analogous to C-G for mixed-integer sets.
- It is straightforward to develop an analog of the rounding principle we used earlier that was geared towards pure integer programs.

**Proposition 4.** *Let* $T = \{x \in \mathbb{Z} \times \mathbb{R}_+ \mid x_1 - x_2 \leq b\}$. *Then the inequality*

$$x_1 - \frac{1}{1 - f_0} x_2 \leq \lfloor b \rfloor.$$

*is valid for* $T$.

- The proof requires exploiting the disjunction

$$x_1 \leq \lfloor b \rfloor \text{ OR } x_1 \geq \lfloor b \rfloor + 1$$

# Mixed Integer Rounding Inequalities

- We can generalize the inequality from the previous slide as follows.

**Proposition 5.** *Let* $T = \{x \in \mathbb{Z}^p_+ \times \mathbb{R}^{n-p}_+ \mid \sum_{j=1}^n a_j x_j \leq b\}$*, where* $a \in \mathbb{Q}^n$ *and* $b \in \mathbb{Q}$*. Then the inequality*

$$\sum_{j=1}^{p}(\lfloor a_j \rfloor + \frac{(f_j - f_0)^+}{1 - f_0})x_j + \frac{1}{1 - f_0} \sum_{p+1 \leq j \leq n : a_j < 0} a_j x_j \leq \lfloor b \rfloor.$$

*is valid for* $T$*, where* $f_j = a_j - \lfloor a_j \rfloor$ *and* $f_0 = b - \lfloor b \rfloor$*.*

- In fact, if $a_j \in \mathbb{Z}$, $gcd\{a_1, \ldots, a_n\} = 1$, and $b \notin \mathbb{Z}$, then the above inequality is facet-inducing for $\text{conv}(T)$.
- The above inequality is called a *mixed integer rounding* (MIR) inequality.
- Its validity can be proved by aggregating the integer and continuous variables, respectively, and applying Proposition **??**.

# 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

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

- Since $k \leq -1$ or $k \geq 0$, we have the disjunction

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

OR

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

# The Gomory Mixed Integer Cut

- Applying Proposition **??**, 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) inequality.
- GMI inequalities 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

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

# GMI vs. MIR

- Although we derived the GMI inequality using a different logic than that which we used for the MIR inequality, they are equivalent.
- Beginning with the inequality

$$\sum_{j=1}^{n} a_j x_j \leq b, \tag{2}$$

we add a slack variables $s = b - a^\top x$ to obtain

$$\sum_{j=1}^{n} a_j x_j + s = b.$$

- Deriving the GMI inequality from this equation and then substituting out the slack variable $s$, we obtain the MIR inequality associated with (**??**).

# Gomory Mixed Integer Cuts from the Tableau

- Let's consider how to generate Gomory mixed integer cuts from the tableau when solving (**??**).
- As before, 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.

$$\max \quad 2x_1 + 5x_2 \tag{3}$$
$$\text{s.t.} \quad 4x_1 + x_2 \leq 28 \tag{4}$$
$$x_1 + 4x_2 \leq 27 \tag{5}$$
$$x_1 - x_2 \leq 1 \tag{6}$$
$$x_1, x_2 \geq 0 \tag{7}$$

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 1: Optimal tableau of the LP relaxation

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

# Example: Gomory Cuts (cont.)

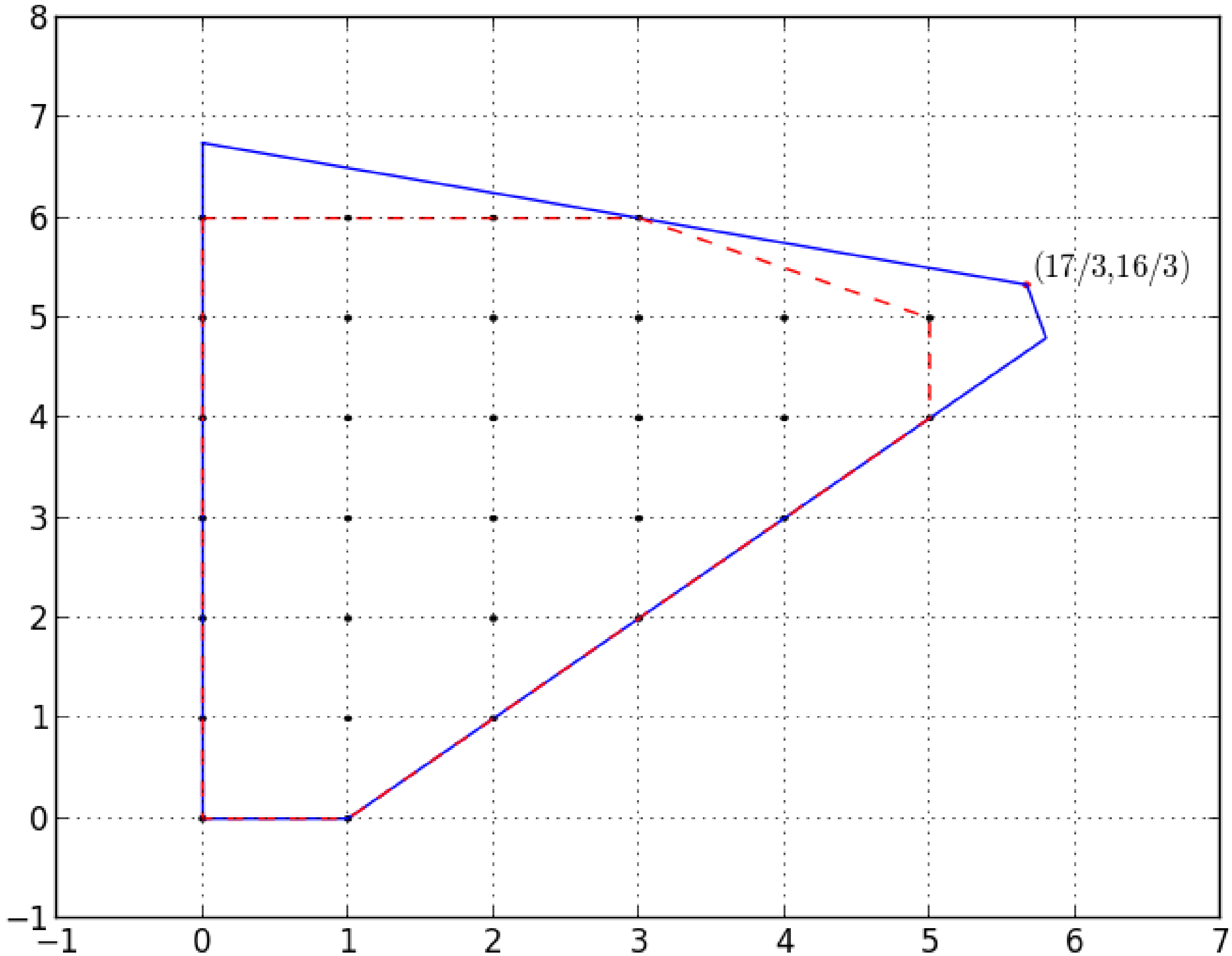



Figure 1: Convex hull of $\mathcal{S}$

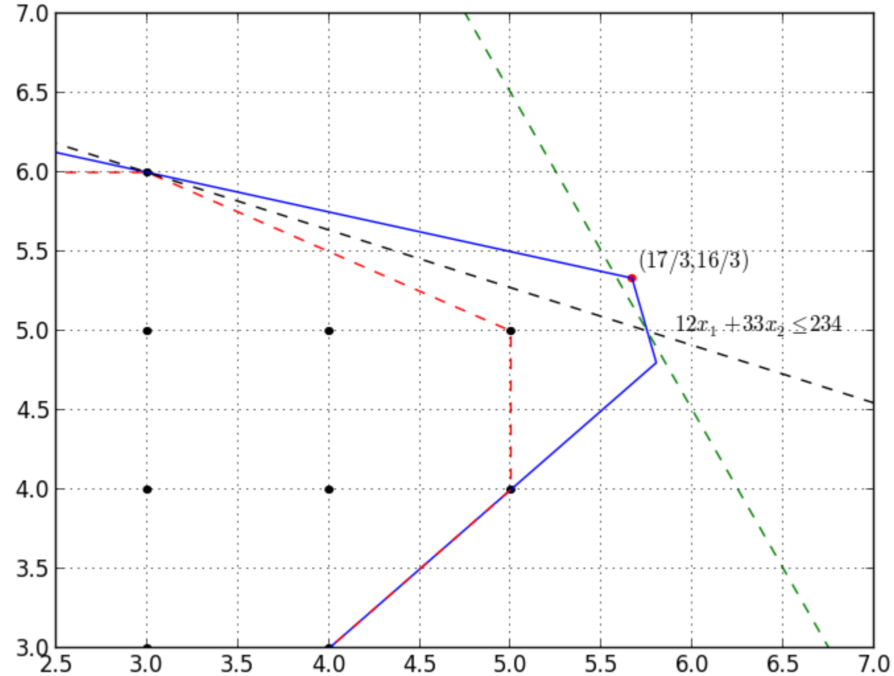
# Example: GMI Cuts versus Gomory Cuts (cont.)

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

$$12x_1 + 33x_2 \leq 234, \qquad \text{(GMI-C1)}$$



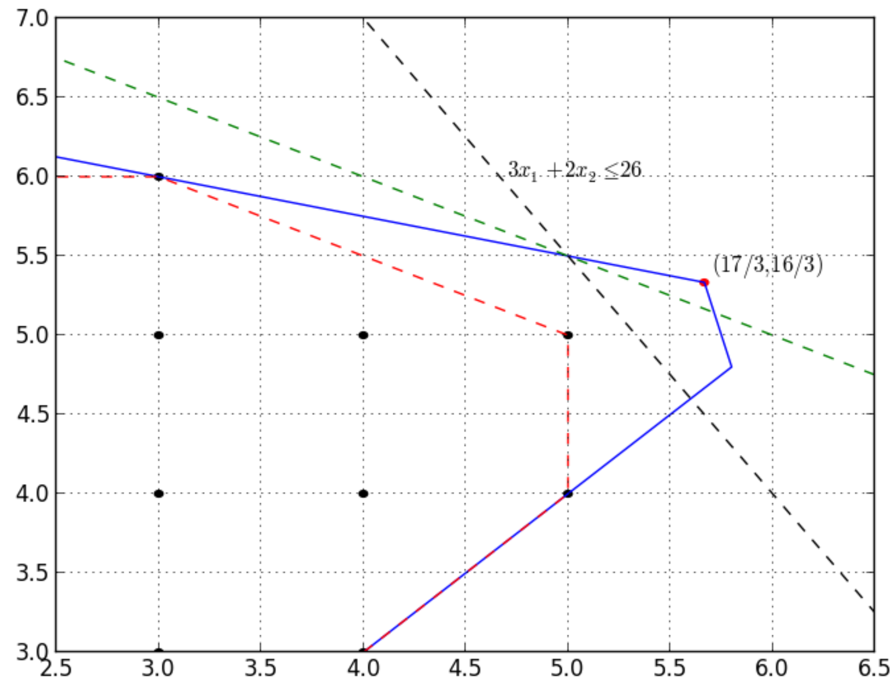
# Example: GMI Cuts versus Gomory Cuts (cont.)

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

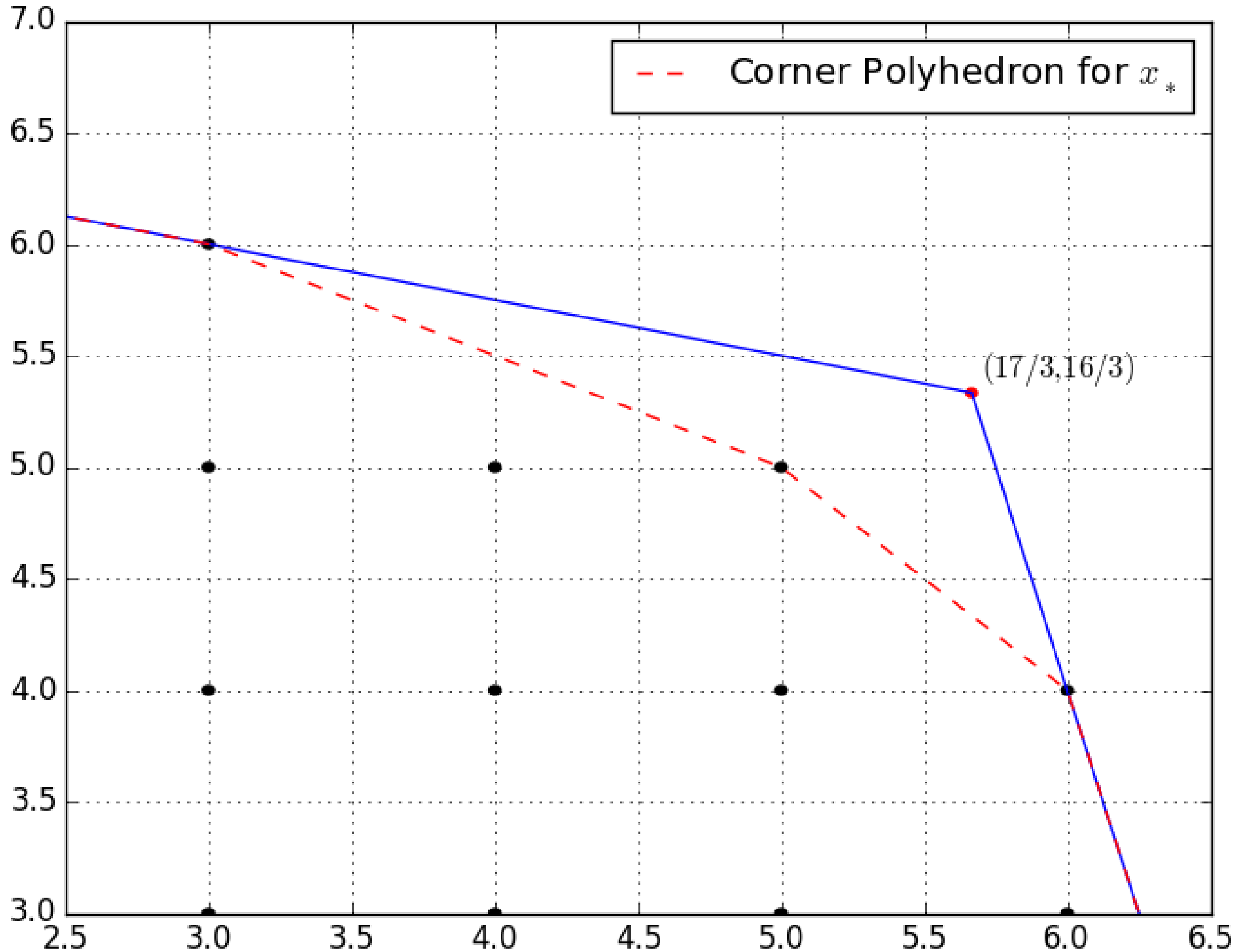



Figure 2: The corner polyhedron associated with the optimal basis of the LP relaxation of the earlier example.

# Intersection Cuts

- A simple way to obtain inequalities valid for the corner polyhedron (and hence the original MILP is as follows.
  - Construct a convex set $C$ whose interior contains the solution $x^*$ associated with the current basis and no other integer points.
  - Determine the points of intersection of each of the extreme rays of the corner polyhedron with the set $C$.
  - The unique hyperplane determined by these points of intersection then separates $x^*$ from the corner polyhedron.
- This is a general paradigm and one can get different classes of valid inequality by choosing the set $C$ in different ways.
- The GMI cut from row $i$ of the tableau is precisely the intersection cut obtained by setting

$$C = \left\{x \in \mathbb{R}^{n+m} \;\middle|\; \lfloor x_j^* \rfloor \leq x_j \leq \lceil x_j^* \rceil \right\},$$

where $x_j$ is the variable that is basic in row $i$.

# Example: GMI Cut as an Intersection Cut

- Figure **??** shows the GMI cut derived from the second row of the tableau in our example as an intersection cut.
- The basic variable in this case is $s_3$.
- In terms of the original variables, we have

$$C = \left\{x \in \mathbb{R}^{n+m} \mid 0 \leq x_1 - x_2 \leq 1\right\}$$

# Example: GMI Cut as an Intersection Cut

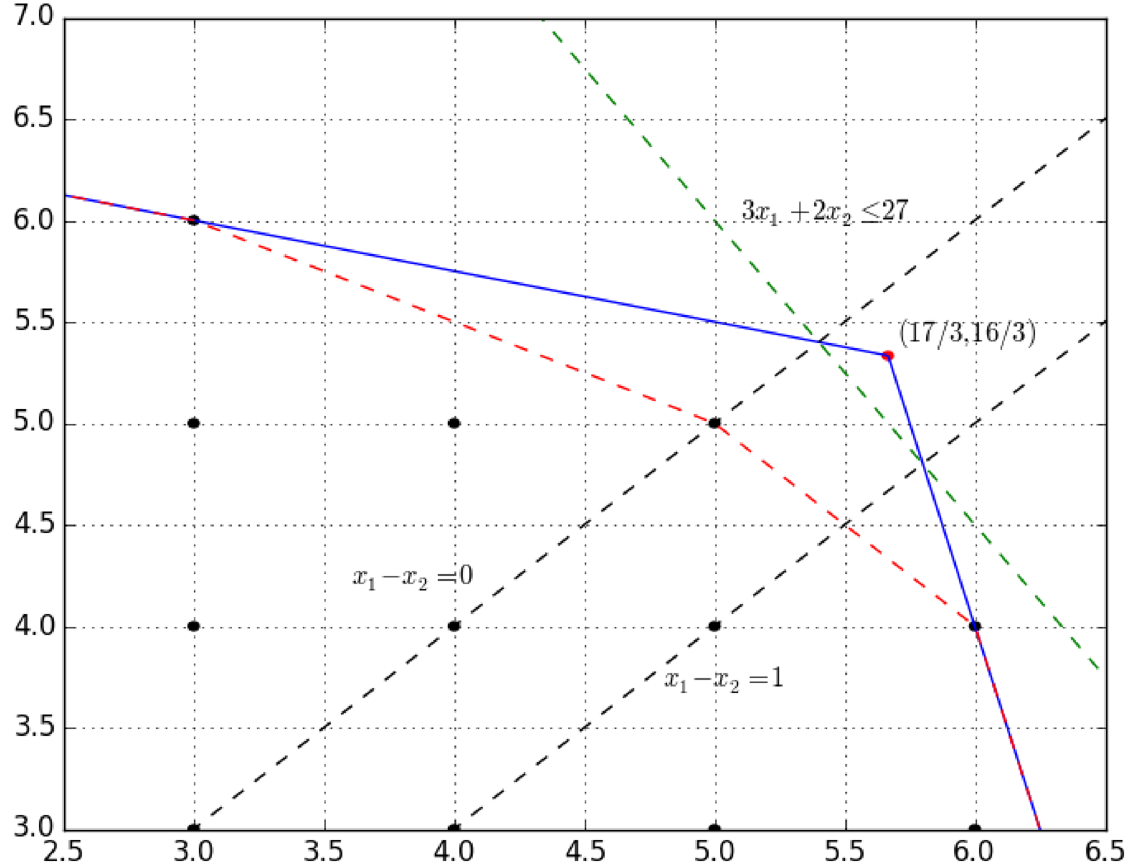



Figure 3: GMI Cut from row 2 as an intersection cut

# GMI Cuts in Practice

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

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

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

# The Polyhedra in Example

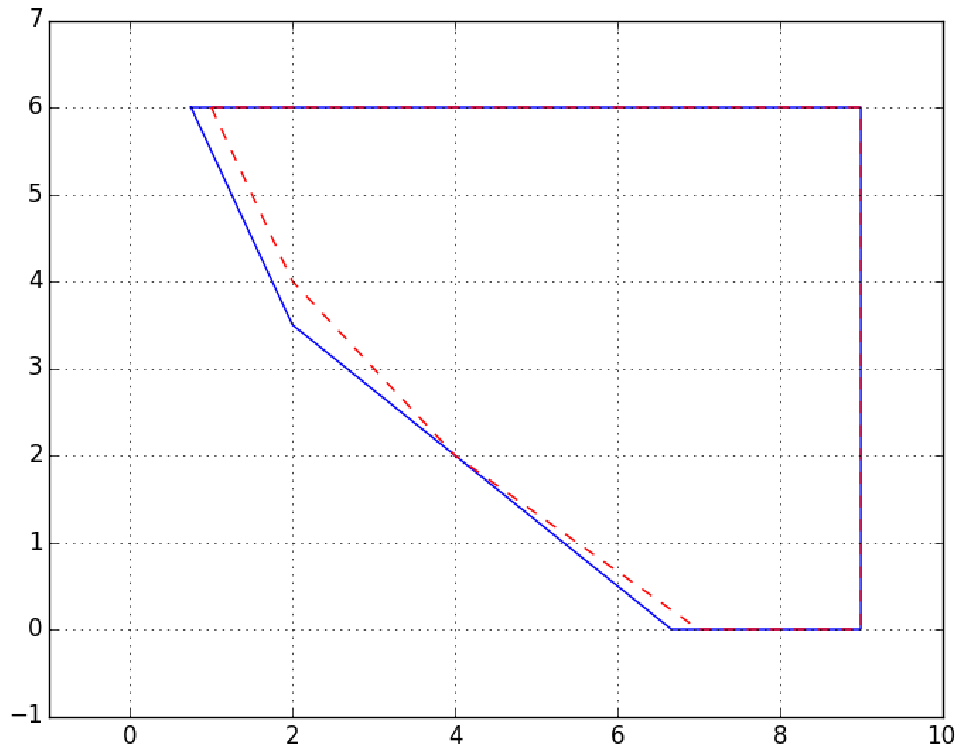



Figure 4: Feasible region of Example MILP

# 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

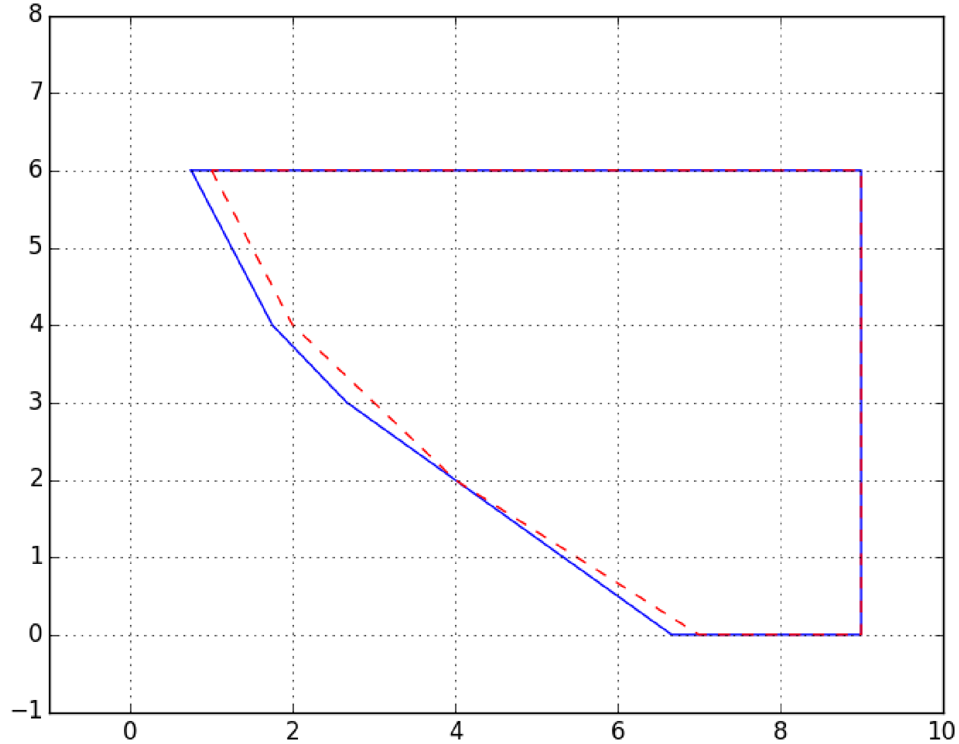



Figure 5: Feasible region of Example MILP after adding cut

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

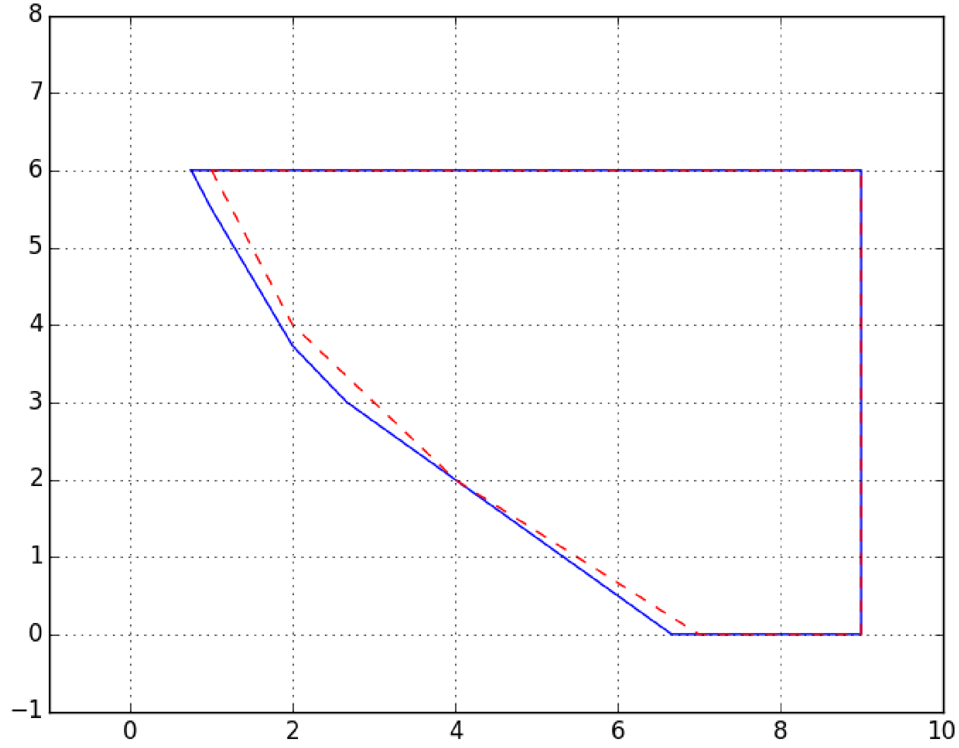



Figure 6: Feasible region of Example MILP after two cuts

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

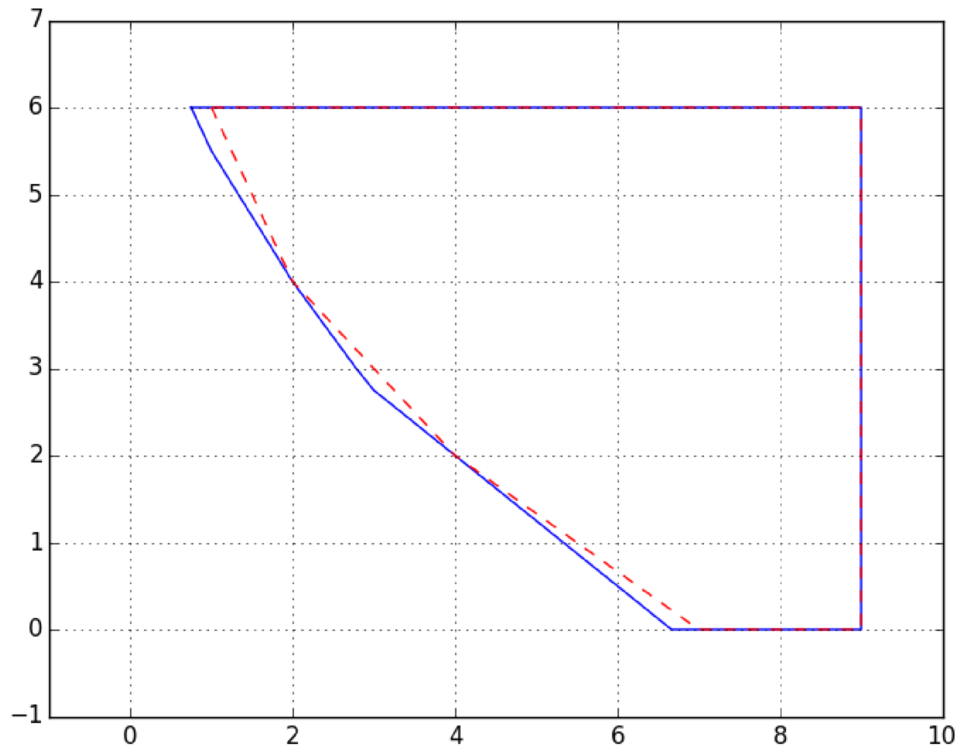



Figure 7: Feasible region of Example MILP after 100 cuts

# 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.

# Lift and Project

- In lift and project, we directly write down the separation problem with respect to a given point and disjunction.
- We will use the variable disjunction on the $j^{th}$ variable.
- We have that $\text{conv}(\mathcal{S}) \subseteq \text{conv}(\mathcal{P}_j^0 \cup \mathcal{P}_j^1)$ where $\mathcal{P}_j^0 = \mathcal{P} \cap \{x \in \mathbb{R}^n \mid x_j \leq \lfloor x_j^* \rfloor\}$ and $\mathcal{P}_j^1 = \mathcal{P} \cap \{x \in \mathbb{R}^n \mid x_j \geq \lceil x_j^* \rceil\}$.
- Applying Proposition **??**, we see that the inequality $(\pi, \pi_0)$ is valid for $\mathcal{P}_j = \text{conv}(\mathcal{P}_j^0 \cup \mathcal{P}_j^1)$ if there exists $u^i \in \mathbb{R}_+^m$, $v^i \in \mathbb{R}_+^n$, and $w^i \in \mathbb{R}_+$ for $i = 0, 1$ such that

$$
\begin{aligned}
\pi &\leq u^0 A + w^0 e_j, \\
\pi &\leq u^1 A - w^1 e_j, \\
\pi^0 &\geq u^0 b + w^0 \lfloor x_j^* \rfloor, \\
\pi^0 &\geq u^1 b - w_1 \lceil x_j^* \rceil,
\end{aligned}
$$

- Notice that this is a set of linear constraints, i.e., we could write an LP to generate constraints based on this disjunction.

# The Cut Generating LP

- This leads to the cut generating LP (CGLP), which generates the most violated inequality valid for $\mathcal{P}_j$.

$$\begin{aligned} \max \quad & \pi\hat{x} - \pi^0 \\ \text{s.t.} \quad & \pi \leq uA + u^0 e_j, \\ & \pi \leq vA - v^0 e_j, \\ & \pi^0 \geq ub + u_0\lfloor x_j^* \rfloor, \\ & \pi^0 \geq vb - v_0\lceil x_j^* \rceil, \\ & \sum_{i=1}^m u_i + u_0 + \sum_{i=1}^m v_i + v_0 = 1 \\ & u, u_0, v, v_0 \geq 0 \end{aligned} \qquad \text{(CGLP)}$$

- The last constraint is for normalization.
- There are a number of alternatives for normalization and the choice does have an impact (see Perregaard).
- This shows that the separation problem for $\mathcal{P}_j$ is polynomially solvable.

# Split Inequalities

- Let $(\alpha, \beta)$ 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 $\operatorname{conv}(\mathcal{P}_1 \cup \mathcal{P}_2)$ 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.

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

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

# Strengthening Lift-and-Project Cuts

- Note that (**??**) only explicitly accounts for the integrality of a single variable.
- We can strengthen the generated cuts using the integrality of the other variables (we consider the pure binary case, but this can be generalized).
- To do this, we simply replace the original coefficients

$$\pi_k = \min\{uA_k, vA_k\} \text{ for } k \neq j$$
$$\pi_j = \min\{uA_j + u^0, vA_j - v^0\}$$

for the integer variables indexed $1 \leq k \leq p$ with

$$\pi_k = \max\{uA_k + u_0\lfloor m_k \rfloor, vA_k - v_0\lceil m_k \rceil\},$$

where

$$m_i = \frac{vA_i - uA_i}{u_0 + v_0}$$

- The proof is to fix the values of $u, v, u_0, v_0$ obtained by solving (**??**) and then find an optimal $(\alpha, \beta)$ in (**??**).

# GMI Cuts vs. Lift-and-Project Cuts

- There is a correspondence between GMI cuts generated from basic solutions of the LP relaxation and strengthened lift-and-project cuts.
  - We use the normalization $\pi_0 \in \{-1, 0, 1\}$ in (**??**).
  - Then each of the former can be derived as the latter from some basic solution to (**??**) (and vice versa, though the relationship is not one-to-one).
- We may be able to get stronger GMI cuts from tableaus other than the one that is optimal to the current LP relaxation.
  - There are lift-and-project cuts that can only be obtained as GMI cuts from an *infeasible tableau*.
  - We may also be able to get stronger cuts from a basic solution that is suboptimal for the LP relaxation.
- By pivoting in the LP relaxation, we can implicitly solve the cut generating LP (see Balas and Perregaard).

# Lift-and-Project Cut as GMI from Infeasible Basis

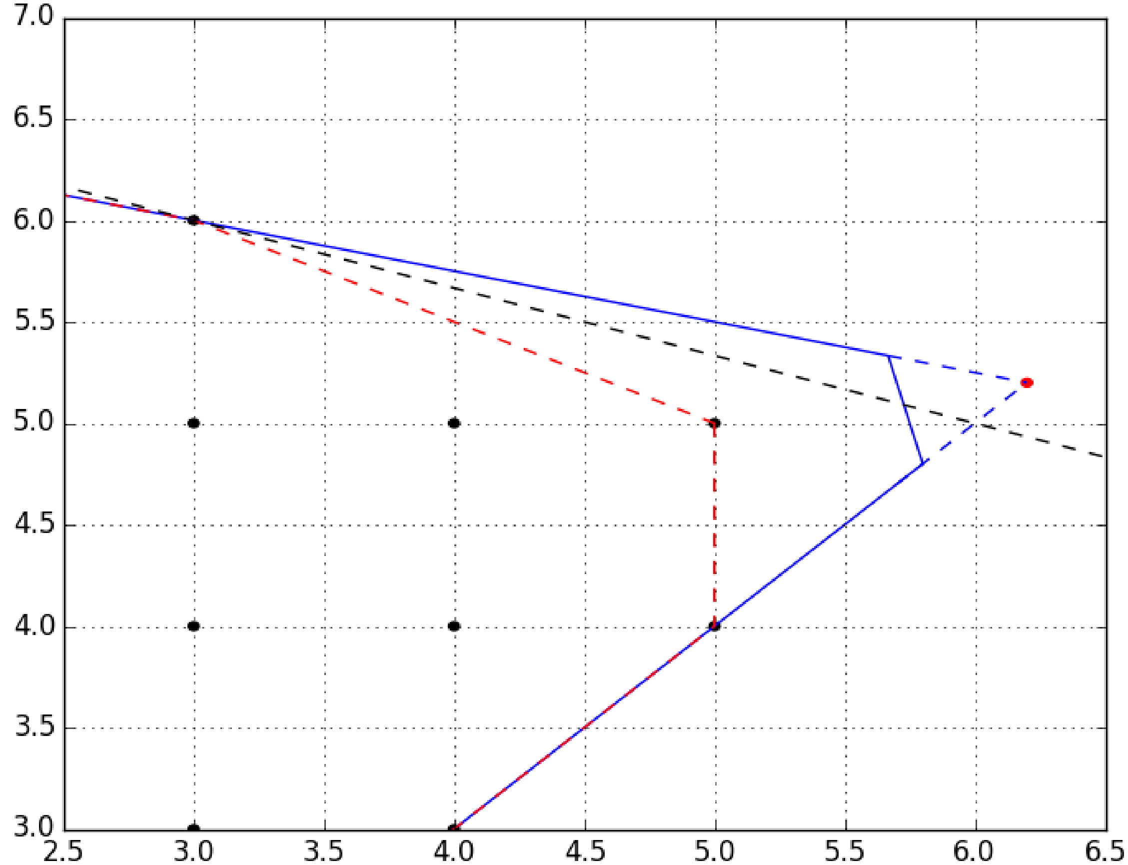



Figure 8: A GMI cut arising from an infeasible basis

# Lift-and-Project Cut as GMI from Alternative Basis

- In our earlier example. the inequality $x_1 \leq 5$ dominates $3x_1 + 2x_2 \leq 26$, but the latter was generated from the current basis.
- With respect to the basic solution $(5.8, 4.8)$, we obtain the cut $x_1 \leq 5$ as a GMI cut.

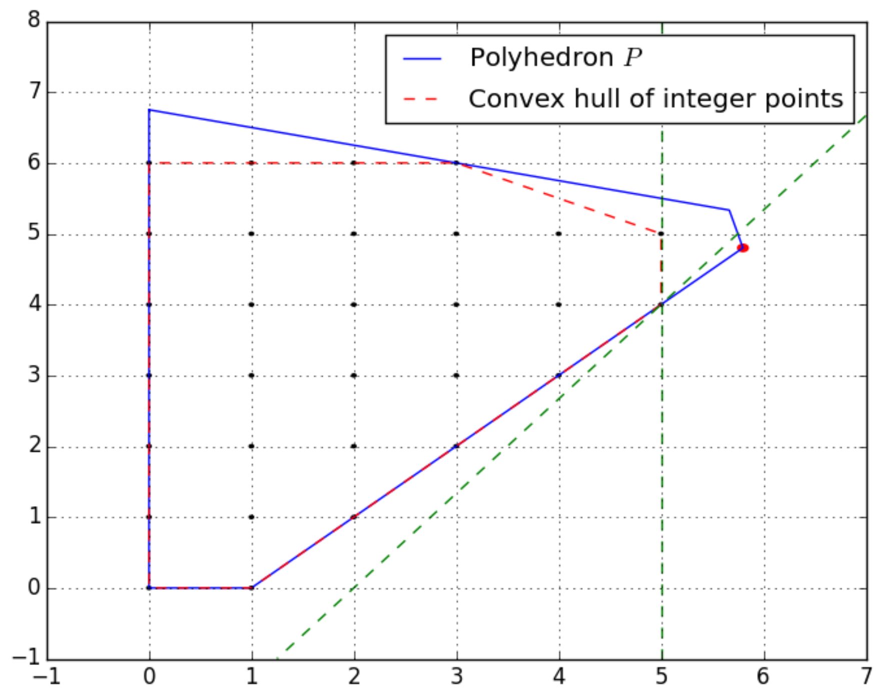



Figure 9: GMI cut arising from an alternative basis

# Another Derivation for Binary Optimization

- Consider the following procedure:
  1. Select $j \in \{1, \ldots, n\}$.
  2. Generate the nonlinear system $x_j(Ax - b) \geq 0$, $(1 - x_j)(Ax - b) \geq 0$.
  3. Linearize the system by substituting $y_i$ for $x_i x_j, i \neq j$, and $x_j$ for $x_j^2$. Call this polyhedron $M_j$.
  4. Project $M_j$ onto the $x$-space.
- In this case, the resulting polyhedron is again $\mathcal{P}_j$.
- This procedure can be strengthened in a number of different ways.

# The Lift-and-Project Closure

- The lift-and-project closure is

$$\mathcal{P}^1 = \cap_{j=1}^n \mathcal{P}_j$$

- We have just shown that optimization over this closure can be accomplished in polynomial time.
- Let $\mathcal{P}^k$ be the lift-and-project closure of $\mathcal{P}^{k-1}$ for $k > 1$.
- The lift-and-project rank of $\mathcal{P}$ is the smallest number $k$ such that $\mathcal{P}^k = \mathrm{conv}(\mathcal{S})$.
- Surprisingly, the lift-and-project rank is bounded by $n$ in the binary and mixed binary case.

# Example: Lift and Project Closure

We consider the polyhedron $\mathcal{P}$ in two dimensions defined by the constraints

$$
\begin{aligned}
-8x_1 + 30x_2 &\leq 115 \\
-2x_1 - 4x_2 &\leq -5 \\
-14x_1 + 8x_2 &\leq 1 \\
2x_1 - 36x_2 &\leq -5 \\
30x_1 - 8x_2 &\leq 191 \\
10x_1 + 10x_2 &\leq 127
\end{aligned}
$$

# Lift-and-Project Closure for Example

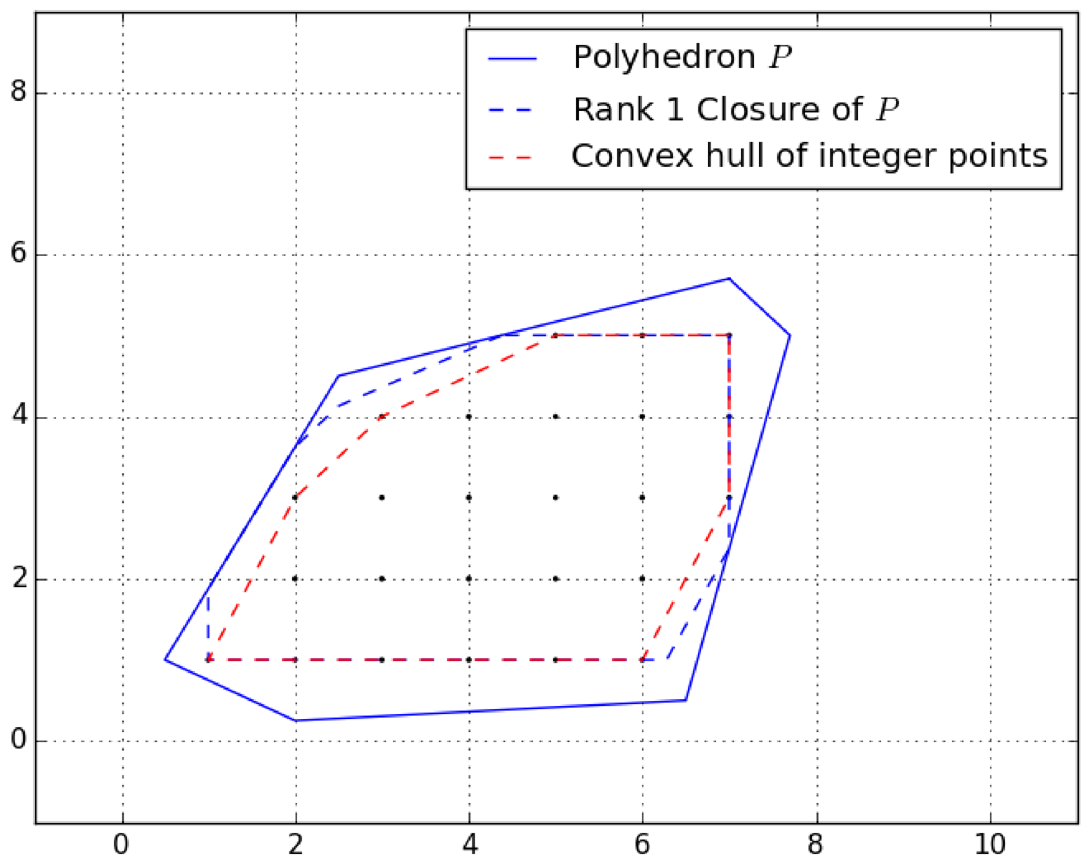



Figure 10: Lift-and-project closure for example

# The GMI closure

- A GMI cut with respect to a polyhedron $\mathcal{P}$ is any cut that can be derived using the GMI procedure starting from any inequality valid for $\mathcal{P}$.
- The GMI closure is obtained by adding all GMI cuts to the description of $\mathcal{P}$.
- The GMI closure is a polyhedron, but in contrast to the lift-and-project closure, optimizing over it is difficult ($\mathcal{NP}$-hard).
  - This seems like a paradox, since we have shown that most-violated GMI cuts are easy to generate.
  - This is only the case, however, for basic solutions to the LP relaxation—separating arbitrary points is difficult in general.
- The *GMI rank* of both valid inequalities and polyhedra can be defined in a fashion similar to that of the C-G rank (more on this later).

# The GMI Closure and the Split Closure

- The *split closure* is the set of points satisfying all possible split cuts and is a polyhedron.
- Every split cut is also a GMI cut and vice versa.
- The split closure and the GMI closure are therefore identical.
- As expected, the GMI cut corresponding to a given split cut is not necessarily one that can be derived from a basic solution to the LP relaxation.
- We can define the *split rank* of an inequality and of a polyhedron as before.
- In the pure integer case, the split rank (and GMI rank) of $\mathcal{P}$ is finite, but it may not be in the mixed case.
- In the mixed binary case, the split rank is bounded by $n$.

# Aside: Selection Criteria

- The criteria by which we select cuts has a big impact on the overall effectiveness.
- We will see later that we in fact need two different kinds of selection criteria: one for generating cuts and one for choosing which cuts to add.
- We typically use bound improvement as a rough criteria when selecting disjunctions for branching, but we often use degree of violation with cuts.
- Why the difference?
- One simple answer is that degree of violation is a linear objective with respect to the cut generating LP.
- Generating cuts according to other criteria seems to be more difficult.
- See
  `http://coral.ie.lehigh.edu/~ted/files/talks/DisjunctionINFORMS12.pdf`
  `http://coral.ie.lehigh.edu/~jeff/mip-2006/posters/Fukasawa.pdf`