Integer Programming
ISE 418

Lecture 15

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.
- "Corner Polyhedra and Intersection Cuts," M. Conforti, G. Cornuejols, and G. Zambelli.
- "Generating Disjunctive Cuts for Mixed Integer Programs," M. Perregaard.
- "On Optimizing Over Lift-And-Project Closures," P. Bonami.
- "A Disjunctive Cutting Plane Procedure for General Mixed-integer Linear Programs," J. Owens and S. Mehrotra.

Split Inequalities

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

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

- Any inequality valid for $\operatorname{conv}(\mathcal{P}_1 \cup \mathcal{P}_2)$ is valid for $\mathcal{S}$ and is called a *split inequality*.
- Chvátal inequalities are exactly the split inequalities for which $\mathcal{P}_2 = \emptyset$.
- The *split closure* of $\mathcal{P}$ is analagous to the Cgvátal closure and is the set described by all split inequalities.
- It is not obvious, but we will see the split closure is a polyhedron.

Deriving Split Cuts

- The cuts we have discussed so far were derived more or less in a single step by choosing a weight vector $u \in \mathbb{R}^m_+$.
- This weight vector combines the inequalities (or equations, if in standard form) into a single new inequality from which cuts can be drived.
- Split inequalities seem to be something more powerful that requires a multi-step process.
  - First, choose a split disjunction (α, β).
  - Then derive an inequality valid for $\mathcal{P}_1$ and an inequality valid for $\mathcal{P}_2$.
  - Finally, combine these inequalities into a single inequality valid for $\mathcal{P}_1 \cup \mathcal{P}_2$.
- We will see, however, that this is not really the case.
- It is possible to derive all non-dominated split cuts by a procedure in which we only need to choose a single weight vector $u \in \mathbb{R}^m_+$.
- To see this requires introducing an even more general kind of disjunctive cut known as an *intersection cut*.

The Corner Polyhedron

- Observe that when separating the solution to the LP relaxation, we only need to consider the constraints that are binding, at least in principle.
- It is only the binding constraints that are used to generate the GMICs introduced in the last lecture, for example.
- Consider a relaxation associated with a basis of the LP relaxation obtained by relaxing the non-negativity constraints on the basic variables.
- This is equivalent to relaxing the non-binding constraints and results in

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

where x_N and s_N are the non-basic variables associated with basis $B \subseteq \{1, \ldots, n\}$.

- The convex hull of $T(B)$ is the so-called *corner polyhedron* associated with the basis B.
- The inequalities we have derived from the tableau are all valid for $\mathrm{conv}(\mathcal{T}(B))$, where B is an optimal basis.

Example: Corner Polyhedron

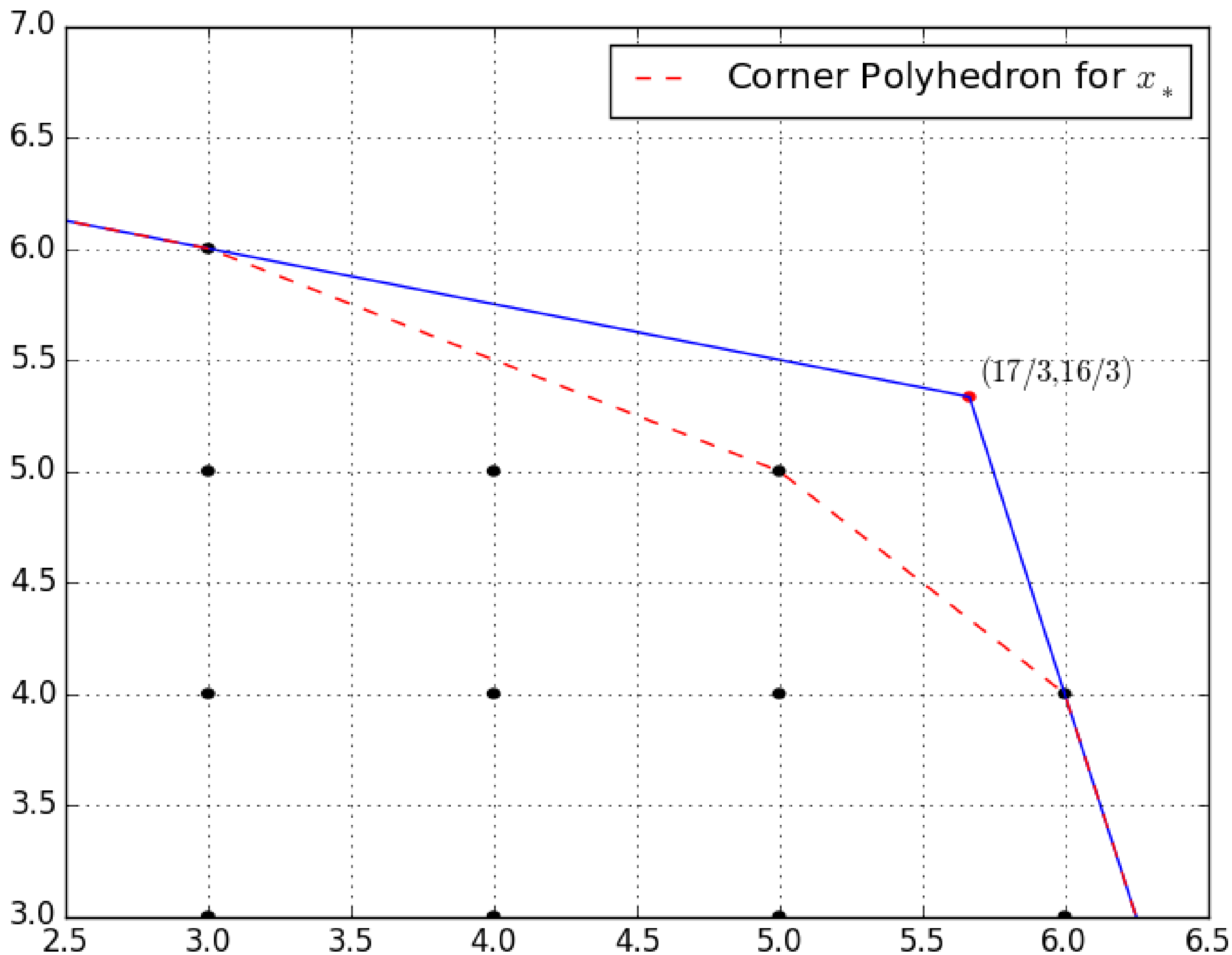

Figure 1: The corner polyhedron associated with the optimal basis of the LP relaxation of our earlier example.

Intersection Cuts

- A simple way to obtain inequalities valid for the corner polyhedron (and hence the original MILP) is as follows.
- First, let $\mathcal{P}(B)$ be the feasible region of the LP relaxation of $T(B)$.
- $\mathcal{P}(B)$ is a socalled *radial cone*, a "translated" cone whose single extreme point is not at the origin, but at another point.
  - Construct a convex set C whose interior contains the solution x^* associated with a basis B and no other integer points.
  - Determine the points of intersection of each of the extreme rays of $\mathcal{P}(B)$ 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.

Intersection Cuts from the Tableau

- Intersection cuts are easy to construct when working with a standard form problem for which we have a tableau associated wth B.
- Note that in this case, the extreme rays of $\mathcal{P}\mathbb{B}$ can be simply constructed from the columns of the tableau.
- Column j of the tableau is $\bar{a}^j = A_B^{-1}A_j$, where A_B is the basis matrix.
- The associated extreme ray r^j of $\mathcal{P}B$ is

$$r_h^j = \begin{cases} -\bar{a}_k^j & \text{if } h \text{ is the variable basic in row } k \\ 1 & \text{if } j = h \\ 0 & \text{otherwise} \end{cases}$$

- The ray r^j is exactly the search direction in the simplex algorithm when pivoting non-basic variable j into the basis.
- Note that these rays are linearly independent and hence, $\mathcal{P}(B)$ is always a $\|N\|$-dimensional polyhedral set.

Recipe for Intersection Cuts

- Given a convex set C, a general recipe for obtaining intersection cuts is as follows.
  - For each $j \in N = \{1, \ldots, n\} \setminus B\}$, compute

$$\alpha_j = \max\{\alpha \geq 0 \mid x^* + \alpha r^j \in C\}$$

  - Note that r^j may belong to the recession cone of C, in which case we take $\alpha_j = +\infty$.
  - Then the intersection cut is

$$\sum_{j \in N} \frac{x_j}{\alpha_j} \geq 1$$

- Intuitively, we can understand that this is the cut we want by noting that it is satisfied at equality for every intersection point.

Example: GMI Cut as an Intersection Cut

- 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.

- We prove this later, for now, we just look at an example.
- Figure 2 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

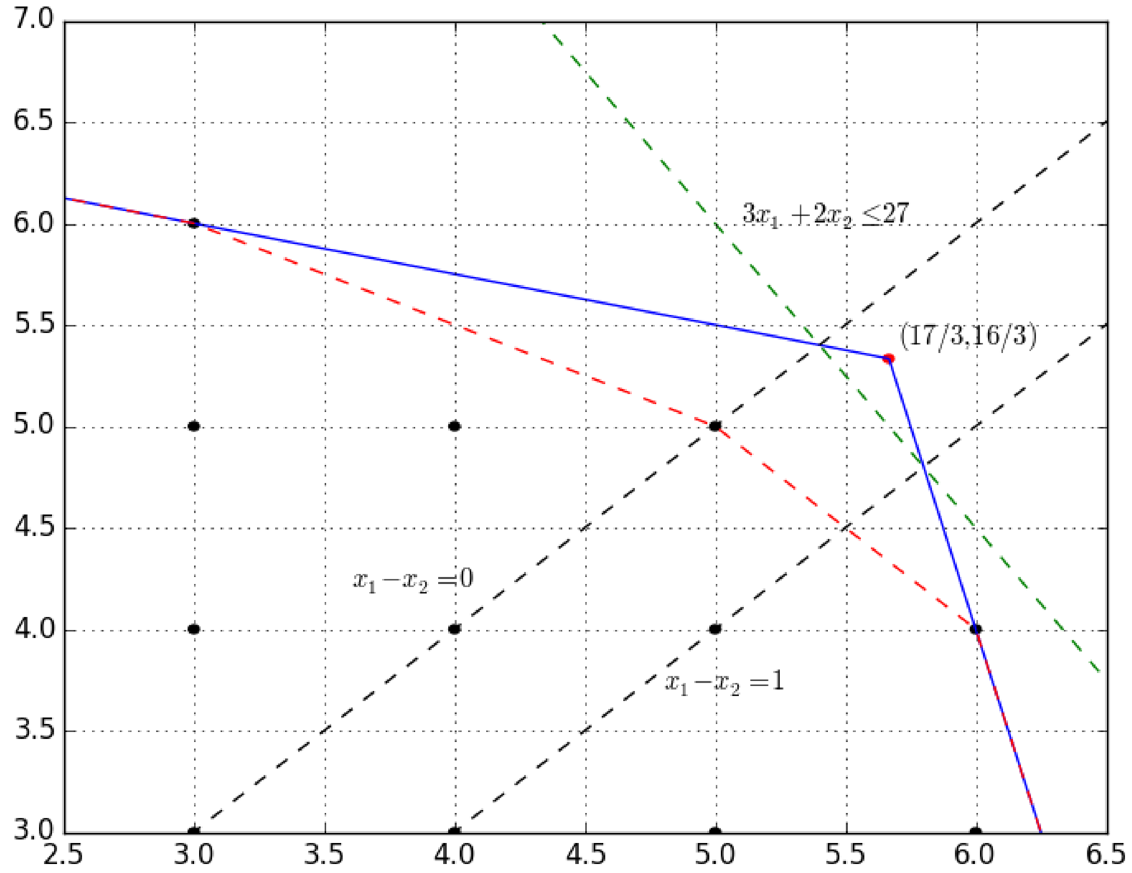

Figure 2: GMI Cut from row 2 as an intersection cut

Constructing Split Inequalities

- The non-dominated split inequalities are those necessary to desribe the split closure.
- Non-dominated split inequalities are obtained by first selecting a basis B (set of n linearly independent constraints indexed by B).
- For a standard form problem, this is equivalent to choosing a set of basic variables.
- Finally, choose $u \in \mathbb{R}^n$ such that
  - $u_i = 0$ for $i \notin B$.
  - $ub \notin \mathbb{Z}$ and $(uA, \lfloor ub \rfloor)$ is a split disjunction.
- Then the corresponding split inequality is

$$\frac{u^+(b - Ax)}{ub - \lfloor ub \rfloor} + \frac{u^-(b - Ax)}{\lceil ub \rceil - ub} \geq 1 \tag{SPLIT}$$

- This is nothing but the intersection cut with respect $T(B)$ and the convex set $\{x \in \mathbb{R}^n \mid \lfloor ub \rfloor \leq uAx \leq \lceil ub \rceil\}$.

Non-dominated Split Cuts are Interssection Cuts

- What we are doing is similar to what we did to get Chvátal inequalities except that we don't require non-negative weights.
- We find a hyperplane with integer coefficients going through some fractional extreme point of $\mathcal{P}$.
- We then "pull the hyperplane apart" into a split disjunction violated by that fractional extreme point.
- With positive weights, $(uA, \lfloor ub \rfloor)$ is a Chvátal inequality.
- Note the inequality again holds at equality for intersection points, e.g.,
  - Let $\bar{x} \in \mathcal{P}_1$ be an intersection point and let $f = ub - \lfloor ub \rfloor$.
  - Let $s_1 = u^+(b - A\bar{x})$, $s_2 = u^-(b - A\bar{x}) \Rightarrow s^1 - s^2 = ub - (uA)^\top \bar{x}$.
  - So $(1-f)s^1 + fs^2 = (1-f)(s^1 - s^2) + s^2 = (1-f)(ub - (uA)^\top \bar{x}) + s^2$
  - Now note that by assumption, $(uA)^\top \bar{x} = \lfloor ub \rfloor \Rightarrow ub - (uA)^\top \bar{x} = f$
  - All but one of the constraints the basis B are also binding at $\bar{x}$.
  - If k is the constraint not binding at $\bar{x}$, then $u_k > 0$, since $uA\bar{x} < ub$.
  - This means $s^2 = 0 \Rightarrow (1-f)s^1 = (1-f)f \Rightarrow \frac{s^1}{f} = \frac{u^+(b - Ax)}{ub - \lfloor ub \rfloor} = 1$

GMI and MIR Cuts are Split Inequalities

- Consider a polyhedron $\mathcal{Q} = \{Ax = b, x \geq 0\}$ in standard form and take $u \in \mathbb{R}^m$, $f_j = (uA)_j - \lfloor (uA)_j \rfloor$, $f_0 = ub - \lfloor ub \rfloor$.
- Then (GMIC) is a split inequality relative to the split disjunction defined by

$$\alpha_j = \begin{cases} \lfloor (uA)_j \rfloor & \text{if } 0 \leq i \leq p \text{ and } f_j \leq f_0 \\ \lceil (uA)_j \rceil & \text{if } 0 \leq i \leq p \text{ and } f_j > f_0 \\ 0 & \text{otherwise} \end{cases}$$

  and $\beta = \lfloor ub \rfloor$.
- Alternatively, an inequality of the form (SPLIT) with respect to the polyhedron

$$\{x \in \mathbb{R}^m \mid Ax \leq b, -Ax \leq -b, -x \leq 0\}$$

  is also equivalent to (GMIC).
- Our proof of validity for MIR cuts showed directly that they are split inequalities.

Split Inequalities are GMI and MIR Cuts

- Let $u \in \mathbb{R}^m$ be such that $uA_I \in \mathbb{Z}$ and $uA_C = 0$.
- The inequality (u^+A, u^+b) is valid for $\mathcal{P}$ and can be re-written as

$$uAx - u^-(b - Ax) \leq ub \ \ \forall x \in \mathcal{P}$$

- But $u^- \in \mathbb{R}^m_+$, so we also have that (u^-A, u^-b) is valid for $\mathcal{P}$ and hence $u^-(b - Ax) \geq 0 \ \ \forall x \in \mathcal{P}$.
- Then, we can apply the same aggregation procedure we used to derive (MIR).
- This results in the inequality

$$uAx - \frac{u^-}{1-f}(b - Ax) \leq \lfloor ub \rfloor$$

which is equivalent to (SPLIT).

- Thus, the MIR, GMI, and split closures are all identical.

Lift and Project

- In lift and project, we formulate the separation problem with respect to a given point and variable disjunction.
- We assume the point to be separated is a basic solution x^B to the LP relaxation with respect to basis B.
- We have that $\mathrm{conv}(\mathcal{S}) \subseteq \mathrm{conv}(\mathcal{P}^B_{0j} \cup \mathcal{P}^B_{1j})$ where $\mathcal{P}^B_{0j} = \mathcal{P} \cap \{x \in \mathbb{R}^n \mid x_j \leq \lfloor x^B_j \rfloor\}$ and $\mathcal{P}^B_{1j} = \mathcal{P} \cap \{x \in \mathbb{R}^n \mid x_j \geq \lceil x^B_j \rceil\}$.
- Applying Proposition 1 of the previous lecture, we see that the inequality (π, π_0) is valid for $\mathcal{P}^B_j = \mathrm{conv}(\mathcal{P}^B_{0j} \cup \mathcal{P}^B_{1j})$ if there exists $u, v \in \mathbb{R}^m_+$, and $u_0, v_0 \in \mathbb{R}_+$ for $i = 0, 1$ such that

$$
\begin{aligned}
\pi &\leq uA + u_0 e_j, \\
\pi &\leq vA - v_0 e_j, \\
\pi_0 &\geq ub + u_0 \lfloor x^B_j \rfloor, \\
\pi_0 &\geq vb - v_0 \lceil x^B_j \rceil,
\end{aligned}
$$

- 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$_j^B$), which generates the most violated inequality valid for $\mathcal{P}_j^B$.

$$
\begin{aligned}
\max \quad & \pi x^* - \pi_0 \\
\text{s.t.} \quad & \pi \le uA + u_0 e_j, \\
& \pi \le vA - v_0 e_j, \\
& \pi_0 \ge ub + u_0 \lfloor x_j^* \rfloor, \\
& \pi_0 \ge vb - v_0 \lceil x_j^* \rceil, \\
& \sum_{i=1}^{m} u_i + u_0 + \sum_{i=1}^{m} v_i + v_0 = 1 \\
& u, u_0, v, v_0 \ge 0
\end{aligned}
\qquad (\text{CGLP}_j^B)
$$

- 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 $\mathcal{P}_j^B$ is polynomially solvable.

Separation Problem for Split Inequalities

- The LP (CGLP$_j^B$) can be generalized straightforwardly to produce the most violated split cut.

$$
\begin{array}{lrl}
\max & \pi x^B - \pi_0 & \\
\text{s.t.} & \pi \leq & uA + u_0\alpha, \\
 & \pi \leq & vA - v_0\alpha, \\
 & \pi_0 \geq & ub + u_0\beta, \\
 & \pi_0 \geq & vb - v_0(\beta + 1), \qquad (\text{SCGLP}^B_{(\alpha,\beta)}) \\
 & \sum_{i=1}^m u_i + u_0 + \sum_{i=1}^m v_i + v_0 = & 1 \\
 & u, u_0, v, v_0 \geq & 0 \\
 & \alpha \in & \mathbb{Z}^n \\
 & \beta \in & \mathbb{Z}
\end{array}
$$

- The separation problem is a mixed integer nonlinear optimization problem, however, and is not easy to solve (unless (α, β) is given).

Strengthening Lift-and-Project Cuts

- Note that (CGLP$_j^B$) 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 (CGLP$_j^B$) and then find an optimal (α, β) in (SCGLP$^B_{(\alpha,\beta)}$).

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 (CGLP$_j^B$).
  - Then each of the former can be derived as the latter from some basic solution to (CGLP$_j^B$) (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

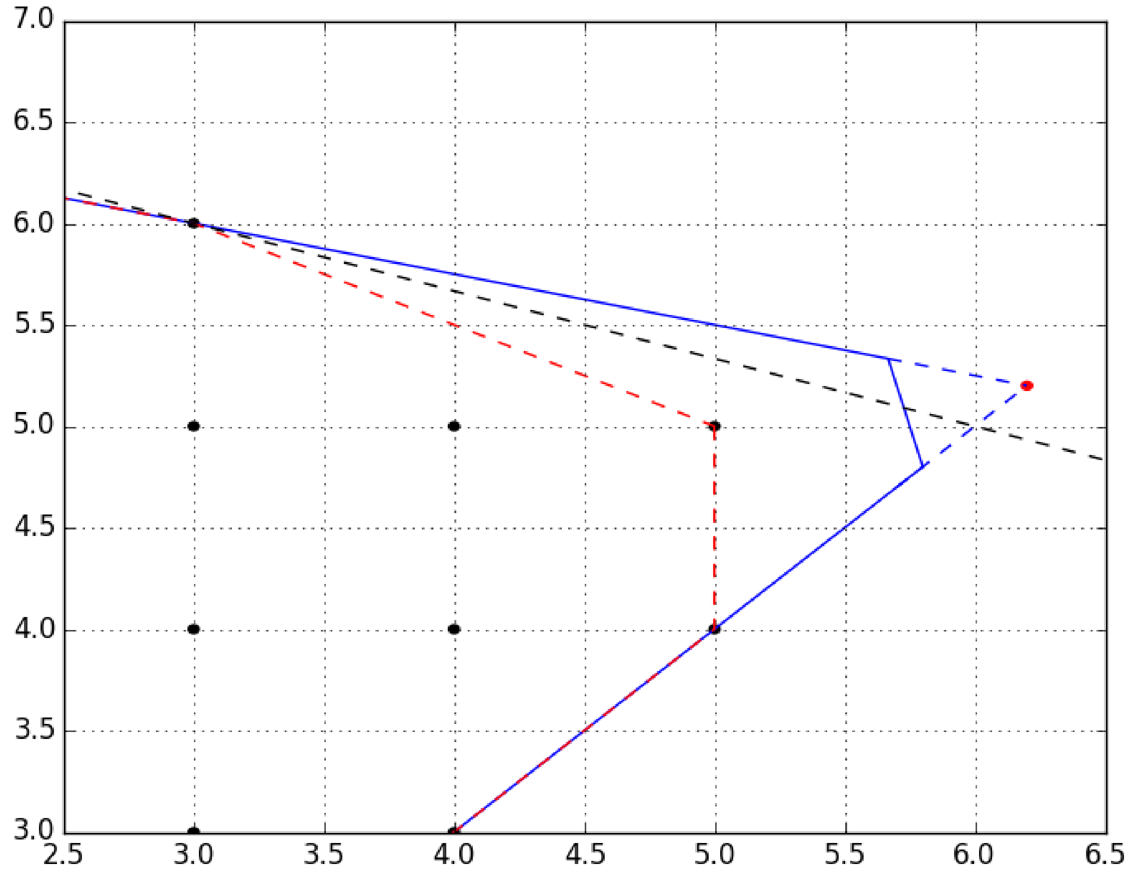

Figure 3: A GMI cut arising from an infeasible basis

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.

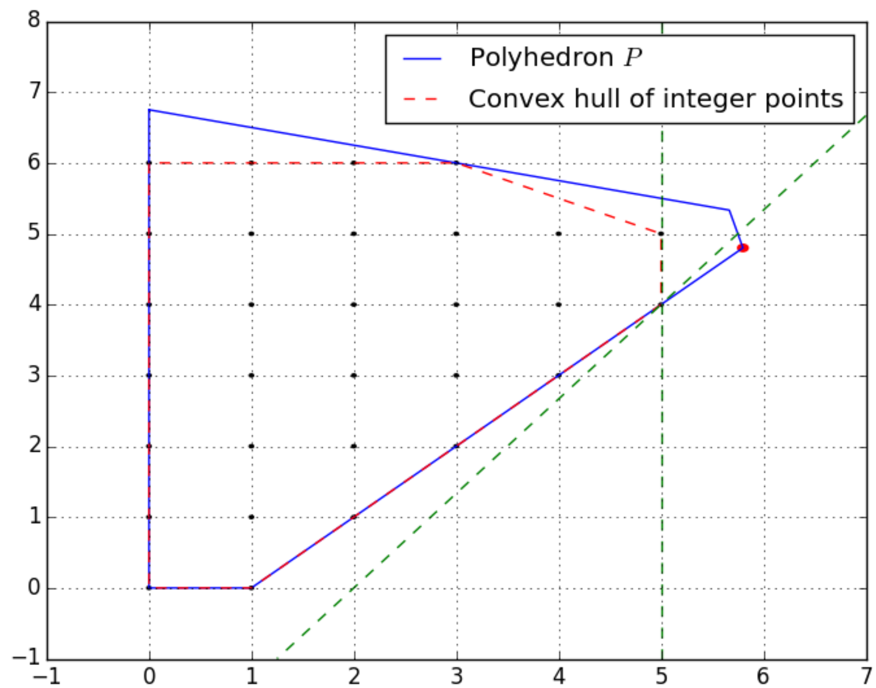

Figure 4: GMI cut arising from an alternative basis

The Lift-and-Project Closure

- The lift-and-project closure is comprised of all inequalities obtained by solving (CGLP$_j^B$) for each basis B of the LP relaxation and each variable j for which $x_j^B \notin \mathbb{Z}$.
- This is equivalent to generating all GMICs corresponding to all rows of all tableaux (feasible or infeasible) of the LP relaxation in which an integer variable is basic with fractional value.
- Note that in the binary case, there is only one possible variable disjunction per variable, so we need only one CGLP per variable.
- Thus, optimization over this closure can be accomplished in polynomial time in the binary case.
- 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

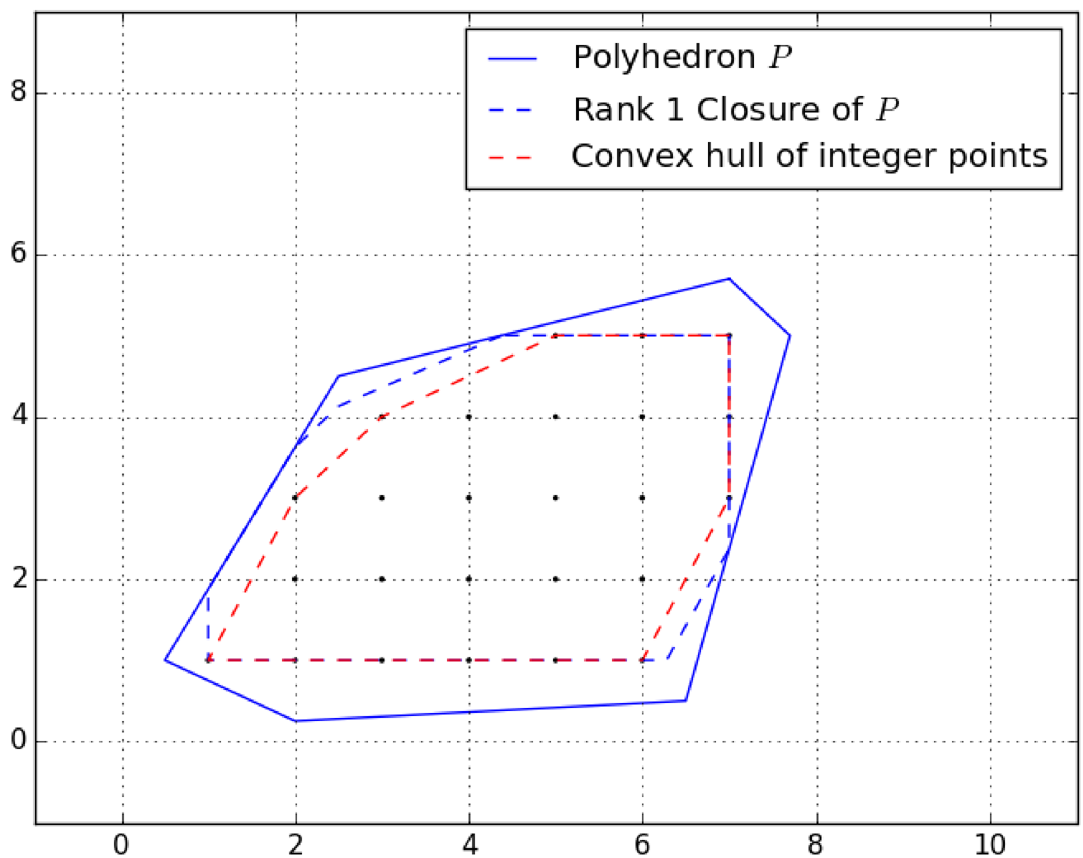

Figure 5: Lift-and-project closure for example

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.
- However, the GMI cut equivalent 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.

Deriving Disjunctive Cuts from Dual Functions

- Recall once more that an inequality (π, π_0) is valid for $\mathrm{conv}(\mathcal{S})$ if

$$\pi_0 \geq F(b),$$

where F is a dual function with respect to the optimization problem

$$\max_{x \in \mathcal{S}} \pi^\top x$$

- When the dual function arises from a (partial) branch-and-bound tree for solving $\max_{x \in \mathcal{S}} \pi^\top x$, we have the general form of a *disjunctive inequality*.
- The simple disjunctive cuts we've talked about so far can be seen as arising from a tree with three nodes in which we branch on exactly one variable.
- We could also easily imagine deriving more complex cuts based on trees with more nodes.
- In fact, any valid disjunction can be used to derive inequalities in this way, whether from a branch-and-bound tree or not.

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`