Cutting Plane Algorithms for Integer Programming

$$\begin{array}{ll} \max & c^T x \\ & Ax \le b \\ & x \ge 0 \\ & x \text{ integer} \end{array} \qquad (IP)$$

$S = \{x \in \mathbb{Z}^n_+ : Ax \le b\}$

$P \doteq \{x \in \mathbb{R}^n_+ : Ax \le b\}$ $\qquad S = P \cap \mathbb{Z}^n.$

Let
$$z_{IP} = \max\{c^T x : x \in S\}$$
$$z_{LP} = \max\{c^T x : x \in P\}.$$

Since $S \subseteq P$, $\quad z_{IP} \le z_{LP}$.

Also, consider the dual LP to $\max\{c^T x : x \in P\}$:

$$\begin{array}{ll} \min & b^T y \\ & A^T y \ge c \\ & y \ge 0 \end{array} \qquad (D)$$

If $\hat{y}$ is feasible in (D) then $b^T \hat{y} \ge z_{LP} \ge z_{IP}$.

So any dual feasible point gives an upper bound on value of

so we can tell how close we are to optimality.

Consider first <u>Gomory's cutting plane algorithm</u>:

Consider equality constrained IP:

$\max \{c^T x : x \in S^e\}$, where $S^e = \{x \in \mathbb{Z}^n_+ : Ax = b\}$

We write problem:

(IP) $\max \{x_0 : (x_0, x) \in S^0\}$, where $S^0 = \{x_0 \in \mathbb{Z}^1, x \in \mathbb{Z}^n_+ : x_0 - c^T x = 0, Ax = b\}$

Suppose an optimal basis for LP relaxation has been obtained, so IP can be written

$$\max \quad x_0$$

$$(\overline{IP}) \qquad x_{B_i} + \sum_{j \in H} \bar{a}_{ij} x_j = \bar{a}_{i0} \qquad i = 0, 1, \ldots, m$$

$$x_{B_0} \in \mathbb{Z}, \quad x_{B_i} \in \mathbb{Z}^1_+ \text{ for } i = 1, \ldots, m, \; x_j \in \mathbb{Z}^1_+ \text{ for } j \in H$$

nonbasic vars

where $x_0 = x_{B_0}$, x_{B_i} are the basic vars $(i = 1, \ldots, m)$, x_j for $j \in H$ are nonbasic vars.

From primal feasibility, $\bar{a}_{i0} \geq 0 \quad i = 1, \ldots, m$

From dual feasibility, $\bar{a}_{0j} \geq 0 \quad$ for $j \in H$

If $\bar{a}_{i0}$ is integer, we're done: optimal soln to LP is opt soln to (IP).

Otherwise: ~~Some~~ $\bar{a}_{i0}$ ~~is~~ non-integral for some i.

~~Proposition~~:
We use the corresponding equation to ~~get~~ generate a cut.

ith row is:

$$x_{B_i} + \sum_{j \in H} \bar{a}_{ij} x_j = \bar{a}_{i0}$$

Let $f_j = \bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor$, $j = 0, \ldots, n$

ith row can be written

$$\underbrace{x_{B_i} + \sum_{j \in H} \lfloor \bar{a}_{ij} \rfloor x_j}_{\text{integral for all feas. solns to IP}} + \underbrace{\sum_{j \in H} f_j x_j}_{\geq 0 \text{ for all feasible solns to IP}} = \lfloor \bar{a}_{i0} \rfloor + \underbrace{f_0}_{\geq 0}$$

So, by C-G type arguments,

$$x_{B_i} + \sum_{j \in H} \lfloor \bar{a}_{ij} \rfloor x_j \leq \lfloor \bar{a}_{i0} \rfloor$$

Separation routine.

or, equivalently: $(*)$ $\sum_{j \in H} f_j x_j \geq f_0$ for any x feasible for (IP).

Note that optimal soln to ~~*~~ L.P violates $(*)$ since $x_j = 0$ $j \in H$ and $f_0 > 0$.

Introduce slack variable x_{n+1}:

$$-\sum_{j \in H} f_j x_j + x_{n+1} = -f_0.$$

Add this constraint to $(\overline{IP})$:

$$\max \quad x_0$$

$$x_{B_i} + \sum_{j \in H} \bar{a}_{ij} x_j = \bar{a}_{i0} \quad i = 0, \ldots, n$$

$$-\sum_{j \in H} f_j x_j + x_{m+1} = -f_0$$

$$x_{B_0} \in \mathbb{Z},\ x_{B_i} \in \mathbb{Z}^1_+ \text{ for } i = 1, \ldots, m+1,\ x_j \in \mathbb{Z}^1_+ \text{ for } j \in H.$$

~~So have primal feasibility, but not dual feasit~~

Don't have primal feasibility, but do have ~~primal~~ dual feasibility.

Reoptimize using dual simplex algorithm.

Example:

$$\begin{array}{llllll} \min & -1.4 & & +1.8x_3 & +2.6x_4 & \\ \text{s.t.} & & x_1 & +1.2x_3 & -0.7x_4 & = 2.5 \\ & & x_2 & -1.4x_3 & +2.3x_4 & = 3.9 \\ & & x_i \geq 0. & & & \end{array}$$

1 − f_{00} since minimizing.

Get cut: Objective: $0.8x_3 + 0.6x_4 \geq 0.4$

Constraints: $0.2x_3 + 0.3x_4 \geq 0.5$

$0.6x_3 + 0.3x_4 \geq 0.9$.

(One feasible integer point: $x = (2, 3, 1, 1)$
There may be others.)

Finite convergence of Gomory's cutting plane algorithm

In order to prove finite convergence, need to ~~define~~ use a **lexicographic** dual simplex algorithm.

Defn Given two vectors g, h, say g is lex. smaller than h, $g \overset{L}{\prec} h$ if $\exists$ index k s.t. $g_i = h_i \quad i = 1, \ldots, k-1$

and $g_k < h_k$.

Eg:
$$\begin{pmatrix}1\\2\\4\\8\end{pmatrix} \overset{L}{\prec} \begin{pmatrix}2\\0\\0\\0\end{pmatrix} \qquad \cancel{\begin{pmatrix}0\\0\\1\\6\end{pmatrix} \overset{L}{\not\prec} \begin{pmatrix}0\\0\\0\\10\end{pmatrix}}$$

$$\begin{pmatrix}0\\0\\2\\6\end{pmatrix} \overset{L}{\prec} \begin{pmatrix}0\\1\\0\\0\end{pmatrix}$$

Will show that optimal primal solutions to successive LP-relaxations strictly decrease lexicographically, provided cuts are generated from the first possible row.

Use "strange" tableau:

$$x_i + \sum_{j \in H} \bar{a}_{ij} x_j = \bar{a}_{i0} \qquad i = 0, \ldots, n \qquad (n = \#\text{ vars in current relaxation})$$

$$x_i \geq 0, \quad i = 1, \ldots, n \qquad H = \text{indices of nonbasic variables}$$

Note that tableau includes rows

$$x_i - x_i = 0 \qquad i \in H.$$

(Corresponds to all reduced costs > 0.)

Assume ~~initially~~ at start of dual simplex that each col of $\bar{A}$ is lexico positive, ie $\bar{a}_{\cdot j} \overset{L}{>} 0$, $j \in H$.

(Easy to ensure this with art. constr.) eg $\sum_{i \in H} x_i \leq M$.

~~Will this that~~ Then using lexico dual simplex, and cutting from first possible row, ensures that we always maintain $\bar{a}_{\cdot j} \overset{L}{>} 0$, $j \in H$.

Also, at every iteration, $\bar{a}_{\cdot 0}$ strictly decreases lexico.

RHS

Min ratio test:

x_p leaving basis, so $\bar{a}_{p0} < 0$.

Standard min ratio test: choose col q which ~~minimizes~~ maximizes

$$\frac{\bar{a}_{0j}}{\bar{a}_{pj}} \quad \text{for } j \in H \text{ with } \bar{a}_{pj} < 0$$

Lexicographic rules affect tie breaks.

If tie, ~~look at~~ ~~seen~~ maximize $\dfrac{\bar{a}_{1j}}{\bar{a}_{pj}}$. Still tied, ~~look at~~ maximize $\dfrac{\bar{a}_{2j}}{\bar{a}_{pj}}$

So rule can be written:

Choose col q which lexico maximizes

$$\frac{1}{\bar{a}_{pj}} \bar{a}_{\cdot j} \quad \text{for } j \in H,\ \bar{a}_{pj} < 0.$$

What happens when we add a cut?

x_0 converges to some limit α_0

If α_0 not integer, when we solve current relaxation we generate cut

$$-\sum_{j\in H} f_{0j}\, x_j + x_{n+1} = -f_{00}.$$

(Recall: $f_{ij} = \bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor$.)

Then at first dual simplex iteration, ~~introduce~~ remove x_{n+1} ~~to~~ from basis.

This reduces ~~x_0 to~~ x_0 to

$$\bar{x}_0 := \alpha_0 - \bar{a}_{0j}\,\frac{f_{00}}{f_{0j}}$$ for some $j \in H$, ~~all $f_{0j} < 0$~~ $f_{0j} > 0$.

$$= \alpha_0 - f_{00}\left(\frac{\bar{a}_{0j}}{f_{0j}}\right)$$

Since $\bar{a}_{\cdot j} \overset{L}{\geq} 0$, we must have $\bar{a}_{0j} \geq 0$

$\therefore\ \bar{a}_{0j} \geq f_{0j} > 0 \qquad \therefore\ \dfrac{\bar{a}_{0j}}{f_{0j}} \geq 1$

$\therefore\ \bar{x}_0 \leq \alpha_0 - f_{00} = \lfloor \alpha_0 \rfloor.$

So obj fn value x_0 decreases to at least $\lfloor \alpha_0 \rfloor$.

So if we always cut on 0th row if possible, x_0 converges finitely to integer & can reduce to next lower integer in finite # its.

Similarly, if we always cut from the first possible row, one of two things happen:

Either i) the ~~cut~~ variable generating the cut decreases to below ~~the~~ the integer part of its value

or ii) an earlier variable decreases.

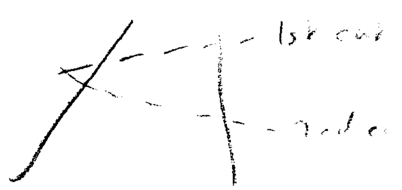
~~In either case, get convergence to a point with one extra~~

in choice of dual simplex pivots

- Theorem (Gomory) Using lexicographic rules, and always deriving the cut from the lowest-indexed non-integer basic var, the cutting plane algorithm converges finitely. //.

In practice convergence is slow; need to add large # cutting planes. Don't get facets in general.

Drop cuts when slack reenters basis; so # vars remains reasonable
Corresponds to cut being superseded



Results from 1990's:

(1) Convergence can be sped up by adding multiple cuts, e.g. add a cut for every fractional basic variable, simultaneously

(Balas, Ceria, Pataki)

(2) ~~Don't need~~ The reduced costs are the dual slacks.

So ~~an interior~~ Gomory cuts from the objective can be derived even with an interior point method.

Also, the projection matrix calculated in an interior point method can be used to approximate $B^{-1}N$, and the limiting values of variables can be somewhat determined.

So Gomory cuts can also be derived from the constraints.

(JEM).

If have vars also ~~embou~~ or upper bounds, then only need one constraint:

min x_{n+1}
s.t. $2x_1 + \dots + 2x_n + x_{n+1} = n$
$0 \le x_i \le 1 \ \forall i$
n odd.

$$
\begin{array}{llll}
\max & x_0 & & \\
\text{s.t.} & x_0 & + x_{n+1} & = 0 \\
& 2x_1 + \dots + 2x_n & + x_{n+1} & = n \\
& x_1 & + s_1 & = 1 \\
& x_2 & + s_2 & = 1 \\
& \vdots & & \\
& x_n & + s_n & = 1 \\
& x_{n+1} & + s_{n+1} & = 1
\end{array}
$$

$x_i \ge 0, \ s_i \ge 0.$

One optimal soln:

$x_1, \dots, x_{\lfloor \frac{n}{2} \rfloor} = 1$, $x_{\lceil \frac{n}{2} \rceil} = \frac{1}{2}$, $x_i = 0$ otherwise, $s_i = 1 - x_i$ ($i = 1, \dots, n+1$)

Tableau, pivot on $x_{\lfloor \frac{n}{2} \rfloor + 1}$ in the long constraint:

$$
\begin{array}{llll}
\max & x_0 & & \\
\text{s.t.} & x_0 & + x_{n+1} & = 0 \\
& \underbrace{x_{\lfloor \frac{n}{2} \rfloor + 1}}_{\text{basic}} + \dots + x_n + \frac{1}{2} x_{n+1} - s_1 - \dots - s_{\lfloor \frac{n}{2} \rfloor} & & = 1/2 \\
& x_1 & + s_1 & = 1 \\
& \vdots & & \\
& x_{\lfloor \frac{n}{2} \rfloor} & + s_{\lfloor \frac{n}{2} \rfloor} & = 1 \\
& \text{basic} & -x_{\lfloor \frac{n}{2} \rfloor + 2} - \dots - x_n - \frac{1}{2} x_{n+1} + s_1 + \dots + s_{\lfloor \frac{n}{2} \rfloor} + s_{\lfloor \frac{n}{2} \rfloor + 1} & = 1/2 \\
& x_{\lfloor \frac{n}{2} \rfloor + 2} & + s_{\lfloor \frac{n}{2} \rfloor + 2} & = 1 \\
& \vdots & & \\
& x_n & + s_n & = 1 \\
& x_{n+1} & + s_{n+1} & = 1 \\
& & \text{basic} &
\end{array}
$$

Gomory cut: $\boxed{x_{n+1} \ge 1}$

How we find a cut is an illustration of a <u>separation routine</u>

<u>Separation problem</u>: Given a point x and a ~~body~~ convex body C, is $x \in C$? If not, give a hyperplane which separates x from C.

So for Gomory:

$C = \text{conv}(S)$.

If $\bar{x}$ optimal for LP, ($\bar{x}$ found using simplex) and $\bar{x}$ not integer, we know $\bar{x} \notin \text{conv}(S)$.

Then the Gomory cut is a hyperplane which separates x from conv(S).

What do we want from a separation routine?

i) Preferably, if $x \notin \text{conv}(S)$, then routine returns a <u>facet</u> of conv(S) which is violated by x.

ii) Separation routine should be fast.