ORIE3310/5310
Lecture 22

April 2013

Recall from last time:

- The cutting stock problem

"Raws" are produced, of width W
There are K different sizes of "finals", of widths $w_1, w_2, \dots, w_K$.
Demand for finals of width w_j is d_j.

We want to satisfy all finals using as few raws as possible.

- 2 Integer Programming Formulations:

	Pros	Cons
Formulation #1:	• $N+NK$ variables • $2N+K$ constraints ∴ not too many variables & constraints.	• The optimal value of its LP-relaxation can be very far from the actual optimal value. ∴ Not informative
Formulation #2:	• M variables • K constraints ∴ not too many constraints but could have a lot of variables	• The optimal value of its LP-relaxation is very close to the actual optimal value. ∴ Very informative.

* let $N :=$ sum of all demands (of finals) $= \sum_{j=1}^{K} d_j$

* let $M :=$ number of "patterns"

- We recall that $a_p = \begin{pmatrix} a_{1p} \\ a_{2p} \\ \vdots \\ a_{Kp} \end{pmatrix}$ is a pattern if

we can cut a_{ip} copies of finals of width w_i, for all $i = 1, \dots, K$, from one raw.

That is, $a_{1p}, \dots, a_{Kp}$ satisfy: $\boxed{\begin{array}{c} a_{1p} w_1 + a_{2p} w_2 + \dots + a_{Kp} w_K \le W \\ a_{ip} \ge 0, \text{ integers}. \end{array}}$ (*)

Suppose that there are M distinct patterns, denoted $a_1, \dots, a_M$, where $a_p = \begin{pmatrix} a_{1p} \\ \vdots \\ a_{Kp} \end{pmatrix}$ for each $p \in \{1, \dots, M\}$,

then the IP formulation (#2) is:

$$\left.\begin{aligned} \text{Min } & \sum_{p=1}^{M} z_p \\ \text{s.t. } & \sum_{p=1}^{M} a_{jp} z_p \geq d_j \quad \forall j \in \{1, \dots, K\}. \\ & z_p \geq 0, \text{ integer} \quad \forall p \in \{1, \dots, M\}. \end{aligned}\right\}$$

where the decision variables z_p = # raws that are cut using the p^{th} pattern.

- How large can M be?

  Suppose that the width of the raws, W, is large compared to the final widths: $w_1, \dots, w_K$. (suppose $w_1 \leq w_2 \leq \dots \leq w_K$)

  Then, the number of different patterns is

  $$\left(\left\lceil \frac{W}{w_K} \right\rceil\right)^k \leq M \leq \left(\left\lceil \frac{W}{w_1} \right\rceil\right)^k$$

  So, $M = O(W^k)$

  $\therefore$ # constraints in Formulation #2 is <u>much</u> larger than $N + NK$

Hence, the LP relaxation:

$$\left.\begin{aligned} \text{Min } & \sum_{p=1}^{M} z_p \\ \text{s.t. } & \sum_{p=1}^{M} a_{jp} z_p \geq d_j \quad j \in \{1, \dots, K\} \\ & z_p \geq 0 \quad p \in \{1, \dots M\} \end{aligned}\right\} (LP)$$

could be too large to be solved efficiently.

- <u>We use a technique called (Delayed) Column Generation</u>

<u>Step 0</u>: Start with just K initial patterns
- Formulate LP relaxation with just these patterns

<u>Ex</u>: (for our numerical example)

$$a_1 = \begin{pmatrix} 3 \\ 0 \\ 0 \end{pmatrix} \quad a_2 = \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix} \quad a_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$(K=3)$

<u>Step 1</u>: Solve the current LP using available patterns.
- Suppose z^* is the opt soln with dual opt soln y^*.
- Using y^*, formulate and solve the <u>subproblem</u>:

$$\left.\begin{aligned} \text{Min } & 1 - \sum_{j=1}^{K} y_j^* a_j \\ \text{s.t. } & \sum_{j=1}^{K} a_j w_j \leq W \\ & a_j \geq 0, \text{ integer} \end{aligned}\right\} (S1)$$

Suppose opt subproblem solution is a^* with optimal value c^*

- If $c^* < 0$, add a^* as a new column $(\widetilde{LP})$, Repeat step 1.
- Else, z^* is optimal for LP

So, solve:

$$\left.\begin{aligned} \text{Min } & z_1 + z_2 + z_3 \\ \text{s.t. } & 3z_1 \geq d_1 \\ & 2z_2 \geq d_2 \\ & z_3 \geq d_3 \\ & z_1, z_2, z_3 \geq 0. \end{aligned}\right\} (\widetilde{LP})$$

Diagram:

We would like to solve the LP.

$$\left.\begin{aligned} &\text{Min} \sum_{p=1}^{M} z_p \\ &\text{s.t.} \sum_{p=1}^{M} a_{jp} z_p \geq d_j \qquad j \in \{1, \dots, k\} \\ &\qquad z_p \geq 0 \qquad\qquad p \in \{1, \dots, M\} \end{aligned}\right\} (LP)$$

But it's too large.

Start with $m < M$ columns. Let $\underline{m = K}$ = # kinds of finals <u>initially</u>.

So, solve $(\tilde{LP})$:

$$\left.\begin{aligned} &\text{Min} \sum_{p=1}^{m} z_p \\ &\text{s.t.} \sum_{p=1}^{m} a_{jp} z_p \geq d_j \qquad j \in \{1, \dots, k\} \\ &\qquad z_p \geq 0 \qquad\qquad p \in \{1, \dots, m\} \end{aligned}\right\} (\tilde{LP})$$

with opt soln z^*, opt dual soln y^*, optimal value $v^* = \sum_{j=1}^{m} z_j^*$

y^* to be used in subproblem's objective function

as long as $v^* < 0$, add a^* as a new column. Add one more variable. $m = m+1$.

Use y^* to solve subproblem:

$$\left.\begin{aligned} &\text{Min } 1 - \sum a_j y_j^* \\ &\text{s.t.} \sum_{j=1}^{K} a_j w_j \leq W \\ &\qquad a_j \geq 0, \underline{\text{integer}} \ \forall j \in \{1, \dots, K\}. \end{aligned}\right\} (Sub)$$

with opt soln a^*, opt value c^*

↳ If $v^* = 0$, then z^* from most recent $(\tilde{LP})$ is opt for (LP).

· <u>Remarks:</u> • The subproblem is a knapsack problem

• Idea on why this method works:

- In $(\widetilde{LP})$, suppose the 10^{th} pattern is not included, this is the same as setting $z_{10}=0$ in original (LP)

∴ basic feasible solutions to $(\widetilde{LP})$ are basic feasible solutions to (LP) with $z_p = 0$ for patterns a_p that are not included.

- Suppose z^* is opt soln for $(\widetilde{LP})$; y^* opt dual soln.
→ In simplex method, we check for new column to enter basis from columns with negative reduced cost.

Suppose a_p is a column, its reduced cost is $\bar{c}_p = c_p - c_B^T B^{-1} a_p$

$$= c_p - (\text{dual soln})^T a_p$$
$$= 1 - y^{*T} a_p$$
$$= 1 - \sum_{j=1}^{K} y_j^* a_{jp}$$

- We know that for columns that are already in $\widetilde{LP}$, $\bar{c}_p \geq 0$ at optimality.
- We want to know if there is a "new" pattern a_p with $\bar{c}_p < 0$.

↓

This is an optimization problem:
Find a pattern a with minimum reduced cost:

$$\text{Min } 1 - \sum_{j=1}^{K} y_j^* a_j$$
$$\text{s.t. } \sum_{j=1}^{K} a_j w_j \leq W$$
$$a_j \geq 0 \text{ int.}$$

Then, if a^*, c^* are the opt solution & opt value, respectively,

if $c^* < 0$, then a^* is a pattern w/ negative reduced cost. So, it can enter basis.
Add to $(\tilde{LP})$ and re-solve.

if $c^* \geq 0$, then there is no pattern that can improve our solution.

so, <u>z^* is optimal for (LP)</u> (not just for $(\tilde{LP})$).

<u>Remark</u>: The optimal value of $(\tilde{LP})$ is always worse than or equal to the opt value of (LP)
because $(\tilde{LP})$ has fewer columns, but the current opt soln of $(\tilde{LP})$ is always feasible for (LP).

<u>Compare to</u>: In cutting planes method,
we add a constraint at each iteration, hoping to obtain an opt solution for the original IP.
At each iteration, the current opt has better opt value than the IP opt value, but the opt soln is not (necessarily) feasible for IP.

- <u>Observe AMPL example</u>:

Iteration 1

Solve ($\tilde{LP}$) Min $z_1 + z_2 + z_3$

$$
\begin{aligned}
3z_1 \quad\quad\quad\quad\quad &\geq d_1 \\
2z_2 \quad\quad &\geq d_2 \\
z_3 &\geq d_3 \\
z_1, z_2, z_3 &\geq 0
\end{aligned}
$$

Opt value: $v^* =$

Opt soln $z^* =$

Dual opt soln: $y^* =$

Solve: Subproblem Min $1 - (y_1^* a_1 + y_2^* a_2 + y_3^* a_3)$

s.t. $3a_1 + 4a_2 + 5.1a_3 \leq 10$

$a_1, a_2, a_3 \geq 0$ integer.

Opt value: $c^* =$

Opt soln: $a^* =$

Iteration 2

Solve ($\tilde{LP}$): Min $z_1 + z_2 + z_3 + z_4$

$$
\begin{aligned}
3z_1 \quad\quad\quad\quad + \quad &\geq d_1 \\
2z_2 \quad\quad + \quad &\geq d_2 \\
z_3 + \quad &\geq d_3 \\
z_1, \ldots, z_4 &\geq 0.
\end{aligned}
$$

Opt value: $v^* =$

Opt value: $z^* =$

Dual solution: $y^* =$

Solve: subproblem: Min $1 - y_1^* a_1 + y_2^* a_2 + y_3^* a_3$

s.t. $3a_1 + 4a_2 + 5.1a_3 \leq 10$

$a_1, a_2, a_3 \geq 0$, int.