Column Generation for Set ~~Covering~~ Partitioning

Eg: Crew scheduling.

Use this approach when we have a very large number of sets (eg, millions). (Hundreds of items.)

$$\min \sum_{j=1}^{n} c_j x_j$$

$$\text{s.t.} \sum_{j: i \in S_j} x_j = 1 \quad \forall i$$

use ~~at least~~ exactly one set that contains item i

(= ~~at least~~ exactly one pairing for each flight leg.)

x_j binary

(S_j corresponds to a column. Eg: S_j is a pairing, then the $i \in S_j$ are the legs covered by that pairing)

LP relaxation:
$$\min \sum_{j=1}^{n} c_j x_j$$

$$\text{s.t.} \sum_{j: i \in S_j} x_j = 1$$

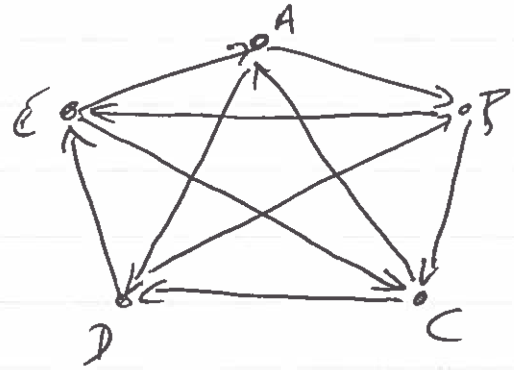
$$x_j \geq 0$$

Dual problem:
$$\max \sum_{i=1}^{m} y_i$$

$$\text{s.t.} \sum_{i \in S_j} y_i \leq c_j \quad \forall \text{ sets } S_j$$

Set partitioning example

Consider five cities & 10 flights:



Can think:
- planes fly ABCDEA, ADBECA
- might take multiple days to make a cycle
- flights are scheduled for each day.

Say we initialize with pairings: ① ABCDEA cost 45
② ADBECA cost 55

Initial LP relaxation:

$$
\begin{array}{llll}
\min & 45x_1 + 55x_2 & & \\
\text{s.t.} & x_1 & = 1 & AB \\
& x_1 & = 1 & BC \\
& x_1 & = 1 & CD \\
& x_1 & = 1 & DE \\
& x_1 & = 1 & EA \\
& \quad x_2 & = 1 & AD \\
& \quad x_2 & = 1 & DB \\
& \quad x_2 & = 1 & BE \\
& \quad x_2 & = 1 & EC \\
& \quad x_2 & = 1 & CA \\
& x_i \geq 0 & &
\end{array}
$$

Dual:

$$
\begin{array}{ll}
\max & \sum y_j \\
\text{s.t.} & y_{AB} + \dots + y_{EA} \leq 45 \\
& y_{AD} + \dots + y_{CA} \leq 55
\end{array}
$$

Multiple dual optimal solns

Eg: $y_{AB} = \dots = y_{EA} = 9$
$y_{AD} = \dots = y_{CA} = 11$

Say pairings available include:

				y sum
③	ABEA	cost	24	29
④	ADEA	cost	23	29
⑤	BCADB	cost	35	41
⑥	ADBEA	cost	43	41

So could add ③, ④, ⑤, but not ⑥

⑥ could be attractive with a different choice of y, eg $y_{EA} = 45, \dots$

Adding ③, ④, ⑤ gives LP relaxation

$$\min\ 45x_1 + 55x_2 + 24x_3 + 23x_4 + 35x_5$$

$$\begin{array}{lllllllll}
\text{s.t.} & x_1 & & + x_3 & & & = 1 & \text{A} \\
& x_1 & & & & + x_5 & = 1 & \text{B} \\
& x_1 & & & & & = 1 & \text{C} \\
& x_1 & & & + x_4 & & = 1 & \text{D} \\
& x_1 & & + x_3 & + x_4 & & = 1 & \text{E} \\
& & x_2 & & + x_4 & + x_5 & = 1 & \text{A} \\
& & x_2 & & & + x_5 & = 1 & \text{D} \\
& & x_2 & + x_3 & & & = 1 & \text{BE} \\
& & x_2 & & & & = 1 & \text{EC} \\
& & x_2 & & & + x_5 & = 1 & \text{CA}
\end{array}$$

$$x_j \geq 0,\ j = 1, \dots, 5$$

Still get primal soln $x_1 = 1,\ x_2 = 1,\ x_3 = x_4 = x_5 = 0$

Dual:

$$\max \sum y_j$$

$$\begin{array}{ll}
\text{st.} & y_{AB} + \dots + y_{EA} \leq 45 \\
& y_{AD} + \dots + y_{CA} \leq 55 \\
& y_{AB} + y_{EA} + y_{BE} \leq 24 \\
& y_{DE} + y_{EA} + y_{AD} \leq 23 \\
& y_{BC} + y_{AD} + y_{DB} + y_{CA} \leq 35
\end{array}$$

suggests pairing CDEC, CDBE

One soln: $y_{CD} = 45,\ y_{EC} = 55$, other $y_i = 0$!

Other solns exist, eg $y_{AB} = 24,\ y_{CD} = 21,\ y_{AD} = 23,\ y_{EC} = 32$, other $y_i = 0$.

Column generation:

Only work with a subset of the sets.

So $x_j = 0$ (forced) for many sets

Thus, when we've got an integer solution to this ~~mod~~ restricted problem, need to check the omitted sets.

Requires checking dual feasibility. PRICING (Checking reduced costs)

~~Since $x_j = 0$ for these extra sets, we have $v_j = 0$ by complementary slackness~~

Thus, we test:

$$\text{Is } \sum_{i \in S_j} y_i \leq c_j \quad \forall \text{ omitted sets } S_j$$

Usually need some heuristic for this, and usually try to find the most violated constraints. For United Airlines, this takes the largest part of ~~the~~ the computational effort.

Add these sets into the primal formulation, and repeat.

Branching:

We may obtain a fractional x.

Could branch on $x_j = 1$ vs $x_j = 0$

But this is far more restrictive than this, so two branches differ greatly in their solvability: $x_j = 1$ branch "easy" (eliminate many items)
$x_j = 0$ branch "hard" (very similar to parent ~~pricing~~ problem).

So look for a set of columns ~~satisfying~~ and two rows i_1, i_2 satisfying

$$0 < \sum_{\substack{j:\, i_1 \in S_j \\ \text{and } i_2 \in S_j}} x_j < 1.$$

~~Set~~ Such a set must exist (Prop 11.3, Wolsey.) follows from being a bfs.

(Constraint matrix:
$$\begin{array}{c} i_1 \\ i_2 \end{array}\begin{bmatrix} 1&1&1&1&1&1&1&0&0&0&0 \\ 0&0&0&0&1&1&1&1&1&1&1 \end{bmatrix} \text{ etc}$$
adding these components of x.)

Not trivial, eg:
$$\begin{bmatrix} 0&1&1&0 \\ 1&0&1&0 \\ 1&1&0&1 \\ 1&1&0&0 \end{bmatrix}$$
x ½ ½ ½ 0

This is bfs. Last two rows don't work, but first two do. Start with $x_1 = \frac{1}{2}$ giving last two rows. This fails, but then this leads to rows 1 & 2 because they don't have x_1 and x_2 in common

Then branch on:

items i_1 and i_2 in the same set vs. items i_1 and i_2 in different sets.

So throw out all columns containing exactly one of i_1, i_2. In ~~the~~ pricing subproblem, add constraint $y_{i_1} = y_{i_2}$ (ie, only consider appropriate S_j.)

So throw out all columns containing both of i_1, i_2. In pricing subproblem, add constraint $y_{i_1} + y_{i_2} \leq 1$ (ie, only consider appropriate S_j.)

Note: these are not the y's from the pricing page, they are binary variables indicating whether an object is in a set.

Note: on both branches, keep columns which contain neither i_1 nor i_2.

Leads to better balance in the subproblems.

Rule due to Ryan & Foster, 1981.

Another way to view the branching decision:

$$\sum_{j: S_j \supseteq \{i_1, i_2\}} x_j = 1 \qquad\qquad \sum_{j: S_j \supseteq \{i_1, i_2\}} x_j = 0$$

items i_1 and i_2 in same set

Here, the linear constraints are $Ax = e$, with

$$A = \begin{array}{c} i_1 \\ i_2 \\ \text{other} \\ \text{items} \end{array} \left[\begin{array}{ccc|ccc} 1 & \cdots & 1 & 0 & \cdots & 0 \\ 1 & \cdots & 1 & 0 & \cdots & 0 \\ \hline & & & & & \\ & & & & & \end{array} \right]$$

$$\underbrace{\qquad\qquad}_{\text{sets containing both } i_1 \text{ and } i_2} \quad \underbrace{\qquad\qquad\qquad}_{\text{sets containing neither } i_1 \text{ nor } i_2}$$

$\downarrow$

So sum of these x_j must equal 1.

items i_1 and i_2 in different sets.

Here, there are <u>no</u> S_j with $S_j \supseteq \{i_1, i_2\}$, so vacuously the sum is 0