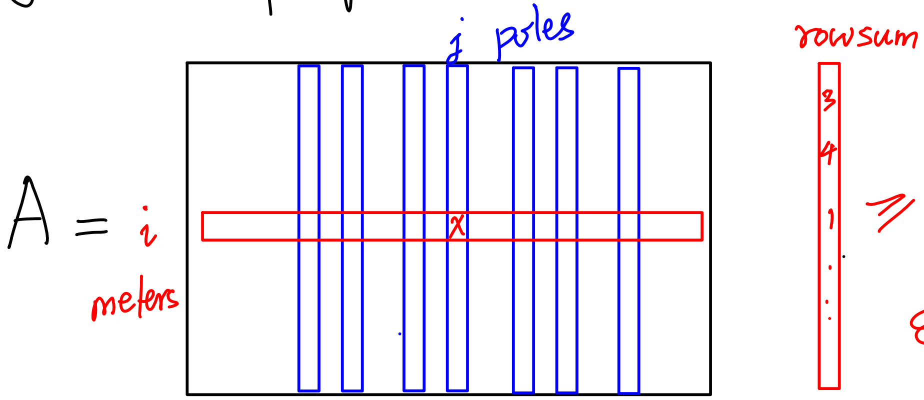
MATH 567: Lecture 28 (04/22/2025)

Today: * About Project 2
* Rangespace reformulation (BR in IP)

Set covering Project

Goal: Ensure every meter is covered, while using as small a # of poles as possible.

$A_{ij} \in \{0,1\}$. A is (very) sparse! You should work with A in sparse format



$\geq \mathbb{1}$ ↳ vector of 1's.

Ensure that the row sum of A is $\geq \mathbb{1}$ (every row sum is ≥ 1) to cover all meters

Identify indices (j) of columns of A, which as a subset, cover all meters.

Perform vector comparisons (row/column) as part of preprocessing & clean-up routines.

Try to avoid **for** loops in Matlab!

One good approach may be to implement all steps correctly first, and then try to optimize in order to meet the run time benchmarks.

Applications of BR in IP

H.W. Lenstra (1983): poly-time algorithm for IP when dimension is fixed. But no implementations are known. There are several similar "theoretical" algorithms, and all of them use BR. Instead, we consider a rather direct application of BR to IP.

Rangespace Reformulation (RSRef)

$$(P)\left\{\begin{array}{l}\bar{b}' \leq A\bar{x} \leq \bar{b}\\ \bar{l} \leq \bar{x} \leq \bar{u}\end{array}\right\} \quad \bar{x} \in \mathbb{Z}^n \quad (IP)$$

IP feasibility problem: $\begin{bmatrix}\bar{b}'\\ \bar{l}\end{bmatrix} \leq B\bar{x} \leq \begin{bmatrix}\bar{b}\\ \bar{u}\end{bmatrix}$, where $B = \begin{bmatrix}A\\ I\end{bmatrix}$.

Given (IP), we apply BR to $B = \begin{bmatrix}A\\ I\end{bmatrix}$.

let $\tilde{B} = BU = \begin{bmatrix}\tilde{A}\\ \tilde{I}\end{bmatrix}$ be LLL-reduced; U is the unimodular matrix for BR.

We consider $(\tilde{P})\left\{\begin{bmatrix}\bar{b}'\\ \bar{l}\end{bmatrix} \leq \tilde{B}\bar{y} \leq \begin{bmatrix}\bar{b}\\ \bar{u}\end{bmatrix}\right\}$ $\bar{y} \in \mathbb{Z}^n$ $(\tilde{IP})$

and try to solve $(\tilde{IP})$ using standard techniques, i.e., using branch-and-cut techniques.

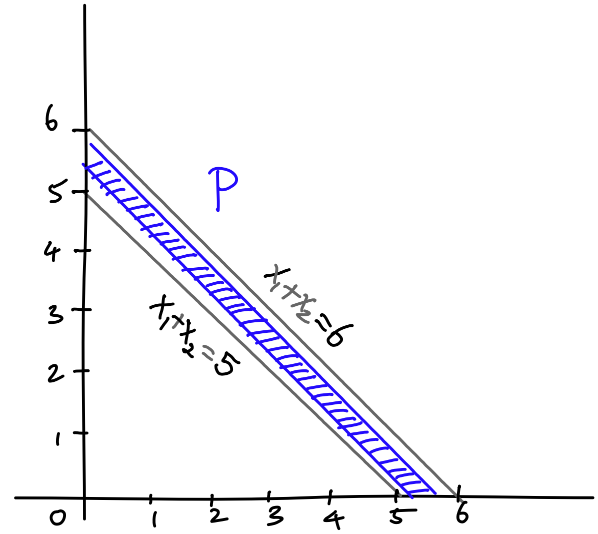
With $\tilde{B} = BU$ we get

$$B\bar{x} = \underline{BU}\,\underline{U^{-1}\bar{x}} = \tilde{B}(U^{-1}\bar{x}) = \tilde{B}\bar{y},$$

and hence $\bar{y} = U^{-1}\bar{x}$. Since U is unimodular, U^{-1} is integral, and hence there is a 1-to-1 correspondence between the integral $\bar{x}$'s and $\bar{y}$'s.

Recall the 2D knapsack problem:

$$\left.\begin{array}{l}(P)\left\{\begin{array}{c}106 \le 21x_1 + 19x_2 \le 113\\ 0 \le x_1, x_2 \le 6\end{array}\right\}\\ \qquad x_1, x_2 \in \mathbb{Z}\end{array}\right\}(KP)$$



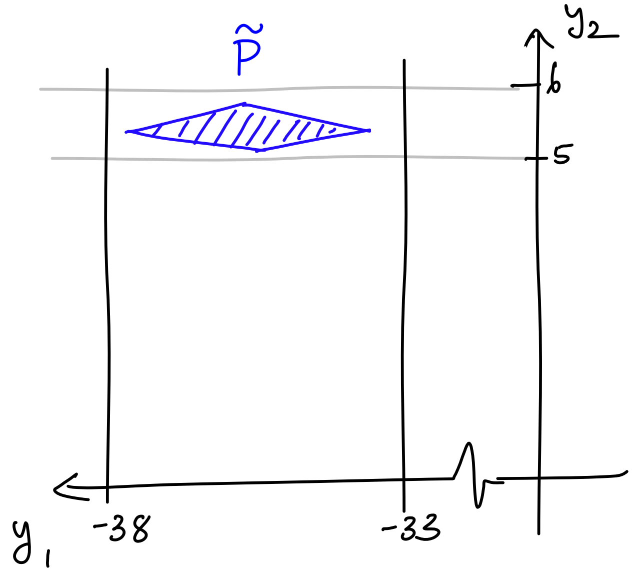
$A = \bar{a}^T = [21 \; 19]$. $\bar{\ell} = \begin{bmatrix}0\\0\end{bmatrix}$, $\bar{u} = \begin{bmatrix}6\\6\end{bmatrix}$.

$B = \begin{bmatrix}21 & 19\\ 1 & 0\\ 0 & 1\end{bmatrix}$. With $U = \begin{bmatrix}-1 & -6\\ 1 & 7\end{bmatrix}$,

$\tilde{B} = BU = \begin{bmatrix}-2 & 7\\ -1 & -6\\ 1 & 7\end{bmatrix}$ is LLL-reduced.

The reformulated problem is

$$\left.\begin{array}{l}(\tilde{P})\left\{\begin{array}{ccccc}106 & \le & -2y_1 + 7y_2 & \le & 113\\ 0 & \le & -y_1 - 6y_2 & \le & 6\\ 0 & \le & y_1 + 7y_2 & \le & 6\end{array}\right\}\\ \qquad y_1, y_2 \in \mathbb{Z}\end{array}\right\}(\tilde{KP})$$



Branching on y_2 solves the problem at the root node itself!

In fact, $\max \{ y_2 \mid \bar{y} \in \tilde{P} \} = 5.94$ and

$\min \{ y_2 \mid \bar{y} \in \tilde{P} \} = 5.04$.

Considering the correspondence between $\bar{y}$ and $\bar{x}$, we get $\bar{y} = U^{-1}\bar{x}$, where $U = \begin{bmatrix} -1 & -6 \\ 1 & 7 \end{bmatrix}$, hence $U^{-1} = \begin{bmatrix} -7 & -6 \\ 1 & 1 \end{bmatrix}$, giving $y_1 = -7x_1 - 6x_2$ and $y_2 = x_1 + x_2$. Hence, branching on y_2 in the reformulation is equivalent to branching on the "good" direction $x_1 + x_2$ in the original problem.

More generally, with $\bar{a} = \bar{p}M + \bar{r}$, we get that branching on y_n in $(\tilde{KP}) \equiv$ branching on $\bar{p}^T\bar{x}$ in (KP).

Notice that BR does not directly "use" the decomposable structure of $\bar{a} = \bar{p}M + \bar{r}$. Still, it recovers the "good" direction $\bar{p}$, and also presents the good direction as the last variable y_n in the reformulation.

This phenomenon generalizes to more than one "good" direction. In fact, there could be a sequence of good directions $\bar{p}_1^T\bar{x}, \bar{p}_2^T\bar{x}, \ldots, \bar{p}_t^T\bar{x}$, and RS Ref will identify them as the collection of variables $\bar{y}_{n-t+1}, \ldots, \bar{y}_n$ in $(\tilde{KP})$. See https://archive.math.wsu.edu/faculty/bkrishna/CKP/ for details and instances.

Cascade knapsack Problem (CKP)

$$9309 \leq \begin{array}{l} 723\,x_1 + 799\,x_2 + 875\,x_3 + 981\,x_4 + 1285\,x_5 + 1361\,x_6 + \\ 1467\,x_7 + 1587\,x_8 + 1847\,x_9 + 1953\,x_{10} + 2029\,x_{11} + 2116\,x_{12} \end{array} \overset{\leq}{\cancel{=}} 9312$$

$$0 \leq x_j \leq 1, \;\; x_j \in \mathbb{Z}, \;\; j = 1, \ldots, 12.$$

The equality version is solved at the root node by Gurobi now, but the inequality version $9309 \leq \bar{a}^T\bar{x} \leq 9312$ takes 539 BB nodes.

The knapsack coefficients decompose as $\bar{a} = \bar{p}_1 M_1 + \bar{p}_2 M_2 + \bar{p}_3 M_3 + \bar{r}$ for

$$\begin{array}{rcl} \mathbf{p}_1 & = & (\;1,\;\; 1,\;\; 1,\;\; 1,\;\; 2,\;\; 2,\;\; 2,\;\; 2,\;\; 3,\;\; 3,\;\; 3,\;\; 3), \\ \mathbf{p}_2 & = & (\;1,\;\; 2,\;\; 3,\;\; 4,\;\; 1,\;\; 2,\;\; 3,\;\; 4,\;\; 1,\;\; 2,\;\; 3,\;\; 4), \\ \mathbf{r} & = & (-1,\;\; 0,\;\; 1,\; -1,\;\; 0,\;\; 1,\; -1,\;\; 0,\;\; 1,\; -1,\;\; 0,\;\; 1). \end{array}$$

and $M_1 = 572$, $M_2 = 97$, $M_3 = 11$.

$$\mathbf{p}_3 = (5,\; 3,\; 1,\; 2,\; 4,\; 2,\; 3,\; 5,\; 3,\; 4,\; 2,\; 1)$$

Branching on $\bar{p}_1^T\bar{x}, \bar{p}_2^T\bar{x}, \bar{p}_3^T\bar{x}$ in that order solves the problem quickly, while branching on x_j's takes exponential # BB nodes.
— see AMPL session.