COURSE #8

Thursday, December 7, 2023 5:53 PM

$$(1)\begin{cases} \min (c^T x) \\ Ax = b \\ x \geq 0 \\ x \in \mathbb{Z} \end{cases} \qquad \begin{cases} \min (c^T x) \\ Ax = b \\ x \geq 0 \end{cases}$$

LP-relaxation of (1)

Q: When does the LP-relaxation of (1) have integral optimum?

Def Matrix A is called Totally Unimodular iff every minor of A has determinant $\pm 1, 0$

A: If matrix A is TU.

SUFFICIENT CONDITIONS FOR A MATRIX TO BE T.U.

SUFF COND #1 If $A = (a_{ij})$ satisfies the following:

(1) $a_{ij} \in \pm 1, 0 \quad \forall i,j$

(2) every column of A has ≤ 2 elements $\neq 0$

(3) Can partition the lines of A into $L = L_1 \cup L_2$ (can be empty!) $(L_1 \cap L_2 = \emptyset)$

s.t. $\forall j \quad \boxed{\sum_{i \in L_1} a_{ij} = \sum_{i \in L_2} a_{ij}}$

$\Downarrow$

$\boxed{A \text{ is T.U.}}$

Jon Lee "Intro to linear optimization" (classroom)

APPLICATIONS OF T.U. to algorithms

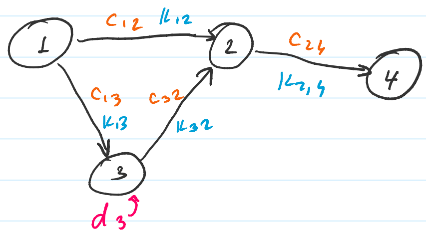
MINIMUM COST FLOW

Given Oriented graph $G = (V, E)$

on each edge (i,j) → capacity $k_{ij} \geq 0$
→ cost $c_{ij} \geq 0$

for every vertex v $\quad d_v \geq 0$ (intuitively the flow that stays in v)

Exp



Flow $\quad F = (f_{ij})_{(i,j)\in E}$

$$0 \leq f_{ij} \leq k_{ij}$$

$\forall v \in V$

$$\boxed{\sum_i f_{iv} - \sum_j f_{vj} = d_v}$$

flow that enters v — flow that leaves v — flow that stays at v

WANT Flow F of minimum cost

$$c(F) = \sum_{(i,j)\in E} c_{ij} f_{ij}$$

$$\min \left(c_{12}x_{12} + c_{13}x_{13} + c_{32}x_{32} + c_{24}x_{24} \right)$$

$$A \rightarrow \boxed{\begin{array}{rrrrl} -x_{12} & -x_{13} & & & = d_1 \\ x_{12} & & +x_{32} & -x_{24} & = d_2 \\ & x_{13} & -x_{32} & & = d_3 \\ & & & x_{24} & = d_4 \end{array}} \rightarrow \text{T.U.}$$

$$\boxed{\begin{array}{l} x_{12} \leq k_{12} \\ x_{13} \leq k_{13} \\ x_{32} \leq k_{32} \\ x_{34} \leq k_{34} \end{array}} \quad \begin{array}{l} y_{12} \\ y_{13} \\ y_{32} \\ y_{34} \end{array}$$

L_1 = all copies

L_2 = empty set

$$\sum_{i\in L_1} x_{ij} = 0$$

$x_{ij} \geq 0$

$(i,j) \quad x_{ij}$

$-x_{ij}$

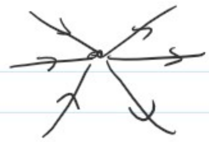
$$\begin{array}{c} \\ \\ \\ \\ \\ \end{array}\begin{array}{c|cccc} & y_{12} & y_{13} & y_{32} & y_{34} \\ \hline A & & & & \\ \hline \begin{smallmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{smallmatrix} & 1 & 0 & 0 & 0 \\ & 0 & 1 & 0 & 0 \\ & 0 & 0 & 1 & 0 \\ & 0 & 0 & 0 & 1 \end{array} \rightarrow \text{T.U.}$$

A is T.U because of suff Condition

But then $\begin{bmatrix} A & 0 \\ B & I \end{bmatrix}$ $\{+B\}$ is T.U.

if all c_{ij}, k_{ij} are integers $\Rightarrow$ OPT cost flow (f_{ij})
d_v has integer values.

MAX-FLOW



$v \neq s, t$ $\{d_v = 0\}$

$0 \leq f_{ij} \leq k_{ij}$ capacity condition keep it

t

$$c_e = \begin{cases} -1 & \text{if } e \text{ enters } t \\ 0 & \text{otherwise.} \end{cases}$$

$$\min\left(\sum_e c_e f_e\right) = \max\left(\sum_{e \to t} f_e\right)$$

CONCLUSION MAX-FLOW special case of MIN-COST flow

$\Downarrow$

$\boxed{\text{simplex can solve MAX-FLOW}}$

(2) $\boxed{\text{SHORTEST PATH PROBLEM}}$

SHORTEST PATH special case of MIN-COST flow

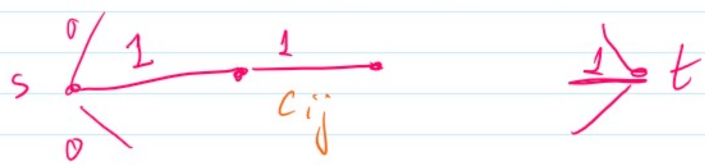
PATH flow

$$d_s = -1$$
$$d_t = 1$$
$$d_v = 0 \quad \forall v \neq s,t.$$

Min-cost flow $\equiv$ flow of unit value from s to t.
of smallest cost.

↓

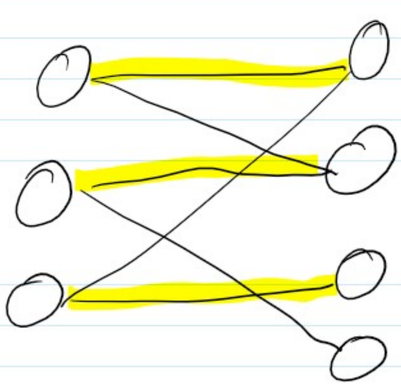
integral



$$c(f) = \sum c_{ij} \underbrace{f_{ij}}_{1} = \sum c_{ij}$$

CONCLUSION Can solve shortest path with SIMPLEX!

③ MATCHING in BIPARTITE GRAPHS



Maximum matching in bipartite graph G

$$x_{ij} = \begin{cases} 1 & (i,j) \in M \\ 0 & \text{otherwise} \end{cases}$$

$$
\underline{\text{ILP}}\begin{cases}
\max\left(\sum_{ij\in E} x_{ij}\right)\\[2mm]
\sum_{k} x_{ik} \le 1 \quad \{\forall i\in L\}\\[2mm]
\sum_{k} x_{kj} \le 1 \quad \{\forall j\in R\}\\[2mm]
x_{ij}\in\{0,1\}
\end{cases}
\qquad \text{LP} \left.\begin{matrix} \\ \\ \\ \\ x_{ij}\ge 0\end{matrix}\right\}
$$

ILP $\Longleftrightarrow$ max matching.

Thm The matrix of LP above is T.U.

CONCLUSION Can SOLVE MAX. MATCHING USING SIMPLEX!

PROOF

$$
\begin{array}{cc|c}
 & & (i,j)\\
L & i & 1\\
\hline
R & j & 1
\end{array}
$$

Cf Suff. Cond $\Rightarrow$ T.U. $\boxtimes$

Def Given ILP

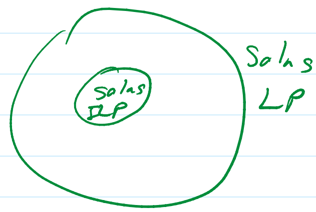
$$
\text{ILP}\begin{cases}\min(c^T x)\\ Ax=b\end{cases} \quad \text{and LP relaxation} \quad \begin{cases}\min(c^T x)\\ Ax=b\end{cases}
$$

$$\text{ILP}\begin{cases}\min(c^T x)\\ Ax = b\\ x \geq 0\\ x \in \mathbb{Z}\end{cases} \quad \text{and} \quad \text{LP relaxation}\begin{cases}\min(c^T x)\\ Ax = b\\ x \geq 0\end{cases}$$

define integrality gap of ILP

$$\boxed{\rho = \frac{OPT(ILP)}{OPT(LP)}} \geq 1$$

$OPT(LP) \leq OPT(ILP)$

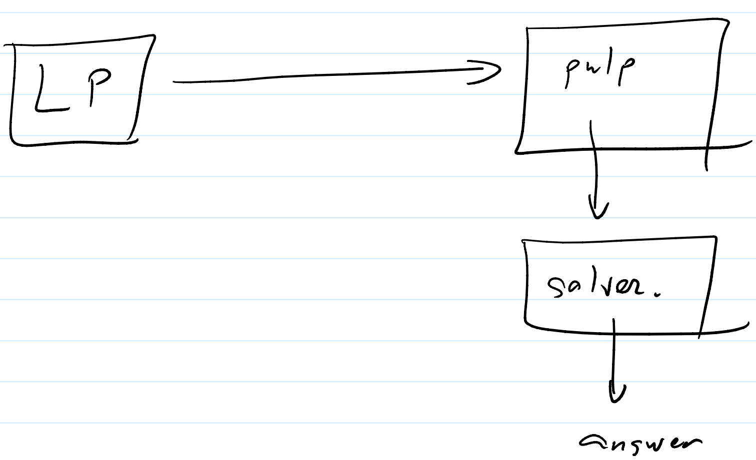


Want formulations of a problem with integrality gap $\simeq 1$

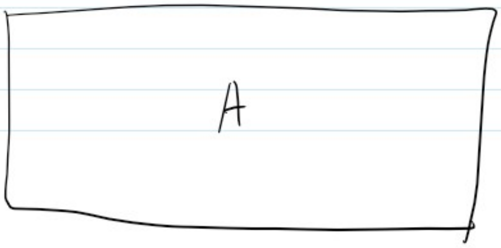
$$Pb \to ILP_1, \quad Pb \to ILP_2$$

Prefer ILP_1 to ILP_2 when $\rho(ILP_1) < \rho(ILP_2)$

Some problems have formulations that are integral but have an exp. large # of equations (variables)



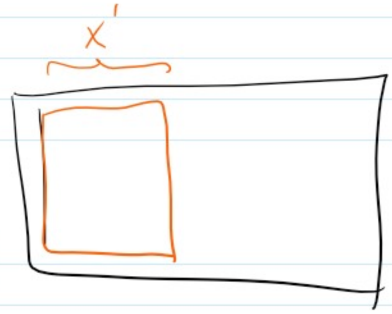
What if I cannot send the pb to the solver (too many equations or too many variables)



COLUMN GENERATION

IDEA ① Start with a Master Pb

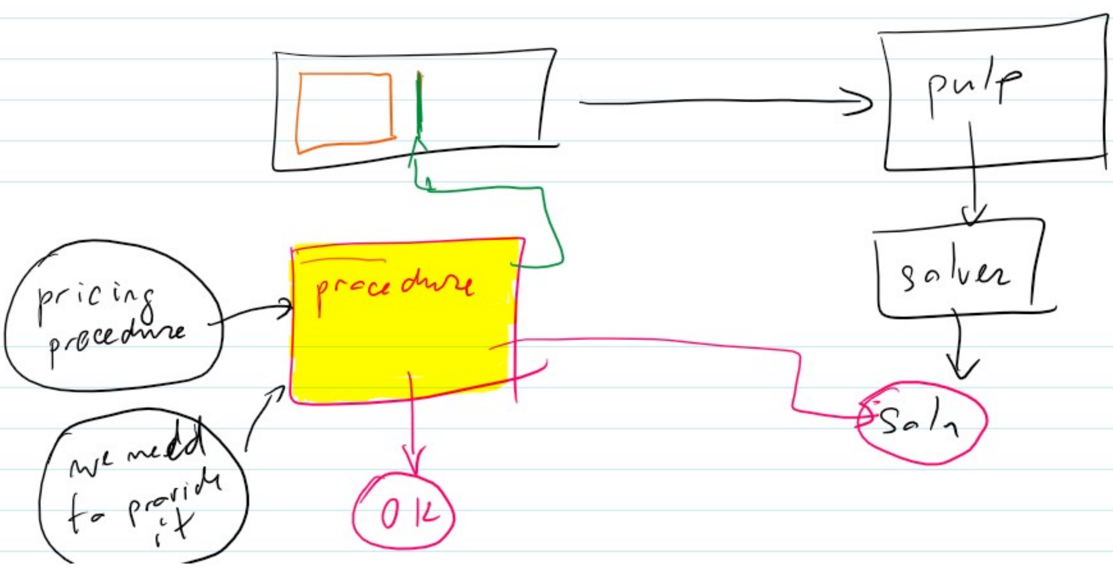
containing only small # of vars



① $\begin{cases} \min(c^T x) \\ Ax = b \\ x \geq 0 \end{cases}$ (2) $\begin{cases} \min(c^T x') \\ A'x' = b \\ x' \geq 0 \end{cases}$

$\begin{cases} x' = \text{OPT soln(2)} \\ x'' = 0 \end{cases}$

feasible soln ①

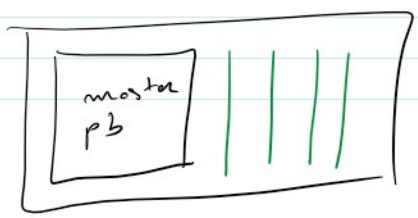


we need to provide it

OK

Soln

callback into SOLVER



JULIA

+

JUMP

provides a uniform interface for callbacks into supported solvers

(1-D) CUTTING STOCK PROBLEM (GILMORE, GOMORY)

$L = 110$ standard wooden boards

l_1 d_1 copies

l_2 d_2 copies

l_m d_m copies

Example

l_i	d_i
20	48
45	35
50	24
55	10
[illegible]	[illegible]

55	10
75	8

GOAL: $\boxed{\text{MINIMIZE total number of standard (big) boards}}$

Pb → $ILP_1 \longrightarrow LP_1$ $\rho(ILP_1)$ very large!

Pb → $ILP_2 \longrightarrow LP_2$ very good $\rho(ILP_2)$

but exp. many variables

requires column generation

Pattern of cutting

$110 = 5 \times 20$ $\{5,0,0,0,0\}$

$110 = 2 \times 45$ $\{0,2,0,0,0\}$

$110 = 2 \times 45 + 1 \times 20$ $[1,2,0,0,0]$

$\boxed{\mathcal{P} = \text{set of all possible cutting patterns}}$

$p \in \mathcal{P}$ x_p = # of boards cut using pattern p.

$p = \{n_{c,1}, \ldots \quad n_{c,m}\}$

$n_{c,i} \in \mathbb{N}$

$$\boxed{\sum_{i=1}^{m} n_{c,i} \, l_i \leq L}$$