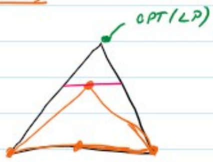
COURSE #12

Friday, January 12, 2024 4:17 PM

Solve MIP (ILP) using branch and cut algorithms

Combine branch & bound with generation of cutting planes

IP $\longrightarrow$ LP relaxation



cutting plane inequality
- fails for the OPT(LP)
- holds for every integral soln

Adding cutting plane it improves LP relaxation

Hope OPT (IP) is a vertex of the improved polyhedron.

YESTERDAY Gomory cutting planes

(2) $\boxed{\text{POLYHEDRAL CUTS (CUTTING PLANES)}}$

facets of the convex hull of feasible integral solns.

$x_1 \le x_2$
$x_1 + x_2 \le 2$ $\Rightarrow$ polyhedral cuts

$x_1 \le 1$ Gomory cut ~~polyhedral cuts~~

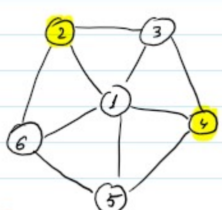
MAXIMUM INDEP. SET

Given Graph $G = (V, E)$

Want indep. set $S \subseteq V$

$|S|$ maximal

$\forall x, y \in S \quad x \not\sim y$



E.g. $S = \{2, 4\}$ $\boxed{OPT = 2}$

$$x_i = \begin{cases} 1 & i \in S \\ 0 & \text{otherwise} \end{cases}$$

$$(IP) \begin{cases} \max (x_1 + x_2 + \dots + x_6) \\ x_1 + x_2 \le 1 \quad (1,2) \\ \vdots \\ x_i + x_j \le 1 \quad (i,j) \in E \\ \vdots \\ x_5 + x_6 \le 1 \quad (5,6) \\ x_1, \dots x_6 \in \{0,1\} \end{cases}$$

$\boxed{OPT(IP) = 2}$ (e.g. $x_2 = 1, x_4 = 1$ all other zeros)

$\Downarrow$

$$(LP) \begin{cases} \max (x_1 + \dots + x_6) \\ x_1 + x_2 \le 1 \\ \vdots \\ x_5 + x_6 \le 1 \\ x_1, \dots x_6 \in [0,1] \end{cases}$$

$\boxed{OPT(LP) = 3}$

$(0.5, 0.5, \dots, 0.5)$

Want cut from soln polyhedron.

Find condition $\rightarrow$ true for every IS .

Find condition → true for every IS.
→ false for $(0.5, \ldots 0.5)$

IDEA Every clique k intersects an indep set S in ≤ 1 vertex

$\forall x, y \in k$
$x \sim y$

$$\boxed{x_1 + x_2 + x_3 \leq 1} \qquad k = \{1,2,3\}$$

Cutting plane – TRUE for all I.S. S
– false for $(0.5, 0.5, 0.5)$

$$(*) \begin{cases} x_1 + x_2 + x_3 \leq 1 \\ \vdots \\ x_1 + x_2 + x_6 \leq 1 \end{cases}$$

polyhedral cuts

$$\boxed{LP_2 = LP_1 + (*)}$$

$OPT(LP_2) = (1/3, 1/3, \ldots 1/3)$

$$\boxed{OPT(LP_2) = 2}$$

③ COVER CUTS

Suppose IP contains an inequality $\boxed{\sum_{j=1}^{m} a_j x_j \leq b}$

$(a_j \geq 0)$

$x_j \in \{0,1\}$

$$(LP) \begin{cases} \sum a_j x_j^* \leq b \\ x_j^* \in \{0,1\} \end{cases}$$

$C \subseteq [n]$ is called a cover iff $\boxed{\sum_{j \in C} a_j > b}$ $\qquad \sum_{j \in C} a_j \geq b+1$

$\Downarrow$

$\exists\, j \in C$ s.t. $x_j = 0$

$\Downarrow$

candidate cover cut → $\boxed{\sum_{j \in C} x_j \leq |C| - 1}$ ①

Stronger version for 1

$$\boxed{E(C) = C \cup \{i : a_i \geq a_j \ \forall j \in C\}}$$

$$\boxed{\sum_{j \in E(C)} x_j \leq |C| - 1} \quad (1')$$

QUESTION – Is (1') cutting plane?

– Is there some C which generates a cutting plane (1') ?

– How do I find such C ?

$\Downarrow$

$\boxed{\text{SOLVE a KNAPSACK problem}}$

Let $x^* = (x_i^*)$ be the optimal soln for the LP

For C to be a cutting plane

I need

$$\sum_{i \in C} x_i^* > |C| - 1$$

$$\Updownarrow$$

$$\boxed{\sum_{i \in C} (1 - x_i^*) < 1}$$

Problem

I don't know C !

Search for C by solving an IP

$$y_j \in \{0,1\} \quad y_j = \begin{cases} 1 & j \in C \\ 0 & j \notin C \end{cases}$$

$$\sum_{i \in C} (1 - x_i^*) = \sum_{j=1}^{n} (1 - x_j^*) y_j$$

$$\sum_{i\in C}(1-x_i^*) = \sum_{j=1}(1-x_j^*)\,y_j$$

$$(IP_*)\quad\begin{cases} \min\left(\sum_{j=1}^{m}(1-x_j^*)\,y_j\right) \\ \sum_{j=1}^{m} a_j y_j \geq b+1 \\ y_j \in \{0,1\} \end{cases}$$

$$\text{solve } (IP_*) \rightarrow y_j^* \begin{cases} OPT < 1 & \underline{\text{cover cut}} \\ OPT \geq 1 & \underline{\text{no cover cut exists for current constraint}} \end{cases}$$

$$(IP_*) \Longleftrightarrow \text{KNAPSACK}$$

$$z_j = 1 - y_j \qquad z_j \in \{0,1\}$$

$$\min\left(\sum (1-x_j^*)(1-z_j)\right)$$

$$\Downarrow$$

profit of object j

$$\begin{cases} \max\left(\sum_{j=1}^{m}(1-x_j^*)\,z_j\right) \\ \sum a_j(1-z_j) \geq b+1 \Longleftrightarrow \overbrace{\sum_{j=1}^{n} a_j z_j \leq \underbrace{\sum_{j=1}^{m} a_j - (b+1)}_{\text{number}}}^{\text{Capacity of knapsack}} \\ z_j \in \{0,1\} \end{cases}$$

$$\sum_{j=1}^{m} (1-x_j^*)\, y_j < 1 \iff \sum (1-x_j^*)(1-z_j) < 1$$

$$\Updownarrow$$

$$\boxed{\sum (1-x_j^*)\, z_j > \sum_{j=1}^{n} (1-x_j^*) - 1}$$

profit
knapsack

TRAVELING SALESMAN (TSP) "comb cuts"

APPLEGATE, BIXBY *THE TRAVELING SALESMAN PROBLEM: A COMPUTATIONAL STUDY*

PRINCETON UNIV. PRESS (2007)

GUROBI

BRANCH AND CUT

different branching strategies

e.g. most infeasible branching

$\{x_j^*\}$ closest to 0.5

SCIP, COIN-OR, ABACUS, (GUROBI CPLEX)

DANTZIG WOLFE DECOMPOSITION (REFORMULATION)

WANT DECOMPOSE $LP \to P_1, P_2 \ldots P_n$

solve these indep.

combine solns. $\to$ soln original pb.

EXAMPLE JOB SCHEDULING ON MULTIPLE MACHINES

J = set of jobs

M = set of machines $\longrightarrow$ $m \in M$

capacity $S_m \geq 0$

$j \in J$, $m \in M$

$w_{j,m}$ = resources taken by j on m

$c_{j,m}$ = cost of running j on m.

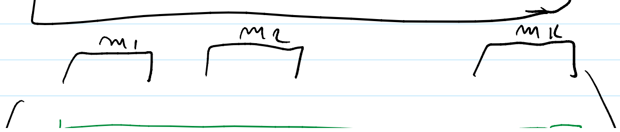
WANT assign each job to a unique machine

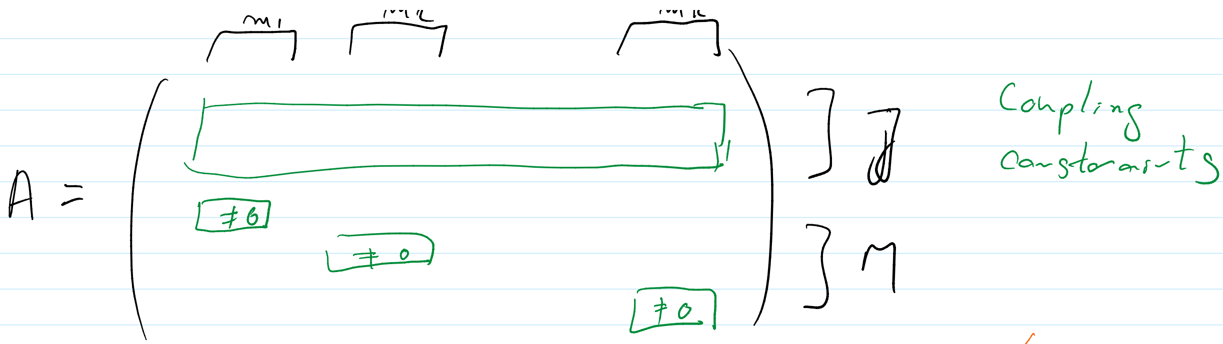
- machine capacity constraints respected
- total cost minimized

$$X_{j,m} = \begin{cases} 1 & j \text{ is assigned to } m \\ 0 & \text{otherwise.} \end{cases}$$

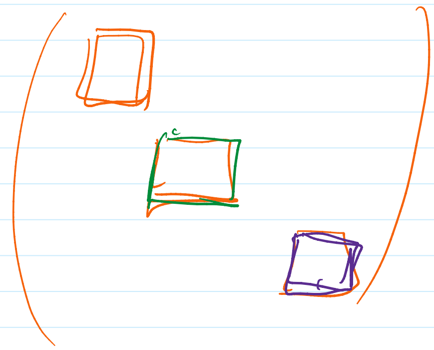
$$(IP) \begin{cases} \min \left(\sum_{j \in J, m \in M} c_{j,m} X_{j,m} \right) \\ \sum_{m \in M} X_{j,m} = 1 \quad (j \in J) & \text{(every job is run on exactly one machine)} \\ \sum_{j \in J} w_{j,m} X_{j,m} \leq S_m \quad (m \in M) & \text{(resource constraint for machine } m\text{)} \\ X_{j,m} \in \{0,1\} \end{cases}$$

$\boxed{\text{LP relaxation terrible}}$

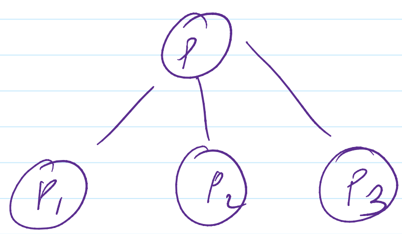




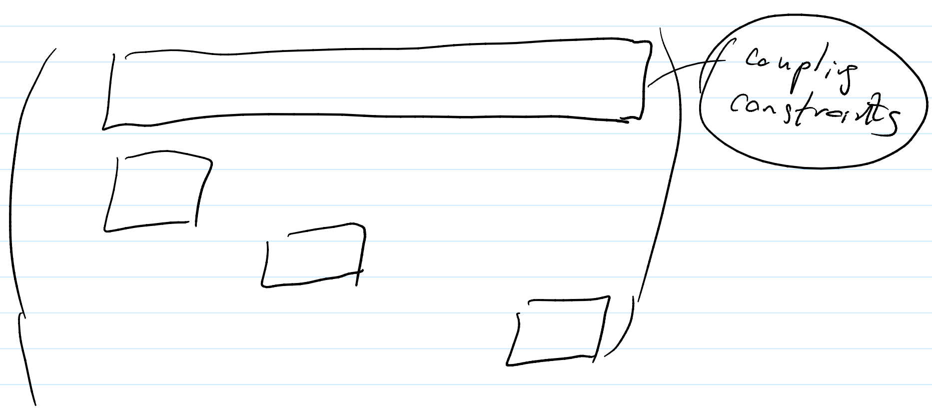
if A was truly diagonal $\longrightarrow$



solving one LP/IP $\longrightarrow$ solving # of indep problems



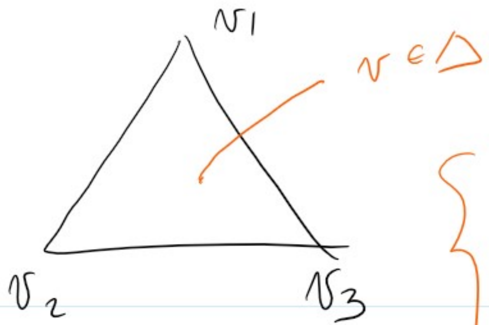
<u>In our case</u> $A =$



<u>Dantzig Wolfe</u> generating good reformulations of the original problem

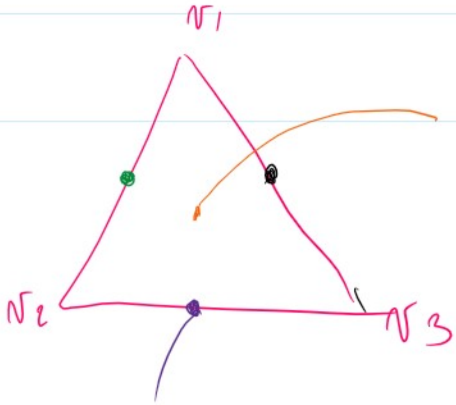
<u>via introducing new variables</u>

X <u>soln polyhedron</u> vertices $v_1, v_2, \ldots v_k$



$$\begin{cases} v = \lambda_1 v_1 + \lambda_2 v_2 + \dots \lambda_n v_n \\ \lambda_1, \dots \lambda_n \geq 0 \\ \lambda_1 + \lambda_2 + \dots + \lambda_n = 1 \end{cases}$$

(convex combination)



$1/3\, v_1 + 1/3\, v_2 + 1/3\, v_3$

$\underline{0.0} \cdot v_1 + \underline{0.5}\, v_2 + \underline{0.5}\, v_3$

$$(1) \begin{cases} \min (c^T x) \\ Ax = b \\ x \in X \longrightarrow \text{polyhedron} \\ x \geq 0 \end{cases}$$

new variables

$$\begin{cases} x = \sum \lambda_i X^i \\ \sum_{i \in V} \lambda_i = 1 \\ \lambda_i \geq 0 \end{cases}$$

D.W reformulation $\Rightarrow$ restating (1) in terms of new variable λ_i'