COURSE #10

Thursday, December 21, 2023 5:58 PM

FIRST WEEK OF CLASSES in 2024:

FRIDAY 16:20 – 17:50

INTEGER LINEAR PROGRAMMING

CAVEAT: ILP is NP-complete.

Algorithms are a form of smart backtracking.

KNAPSACK PROBLEM

NP-complete!

$$\begin{cases} \max(p_1 x_1 + p_2 x_2 + \dots + p_n x_n) \\ w_1 x_1 + w_2 x_2 + \dots + w_n x_n \le k \\ x_i \in \{0,1\} \end{cases}$$

k → knapsack capacity

n objects   object i: profit p_i
weight w_i

$$x_i = \begin{cases} 1 & \text{if we take object } i \\ 0 & \text{otherwise} \end{cases}$$

Exp
$$\begin{cases} \max(45x_1 + 48x_2 + 35x_3) \\ 5x_1 + 8x_2 + 3x_3 \le 10 \\ x_1, x_2, x_3 \in \{0,1\} \end{cases}$$

$\boxed{5}$ 45 ✓

$\boxed{8}$ 48

$\boxed{3}$ 35 ✓

OPT $x_1 = 1, x_2 = 0, x_3 = 1$
OPT(VALUE) = 80

LP Relaxation

LP
$$\begin{cases} \max(45x_1 + 48x_2 + 35x_3) \\ 5x_1 + 8x_2 + 3x_3 \le 10 \\ x_1, x_2, x_3 \in [0,1] \end{cases}$$

$45/5 = 9$
$48/8 = 6$
$35/3 = 11.66$

$\boxed{92 = OPT(LP) \geq OPT(IP) = 80}$

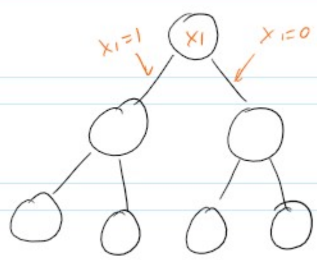
fractional knapsack

greedy
- Sort objects by profit/gram
- take objects in decreasing order of profit/gram until knapsack full
- You may have to cut the last object

$$\begin{cases} x_3 = 1 \\ x_2 = 0.25 \\ x_1 = 1 \end{cases}$$

not integral!

BRANCH AND BOUND ALGORITHM



we would like to prune the backtrack tree

IDEA   Use at each node of the tree an estimate for the optimum.

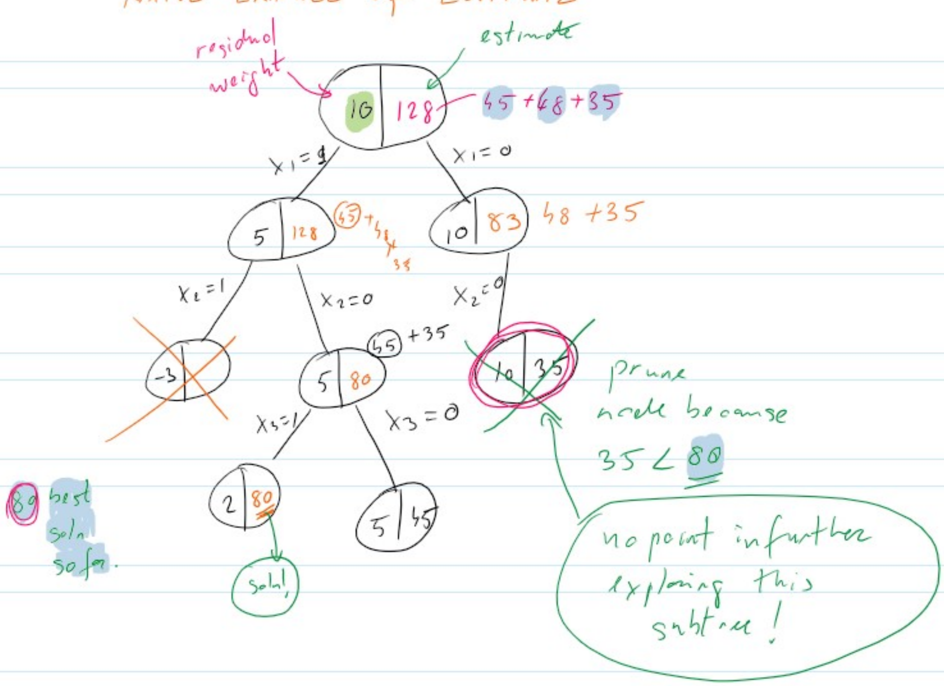
max {OPT ≤ estimate}

How to prune tree using estimate

NAIVE EXAMPLE OF ESTIMATE

node   estimate

NAIVE EXAMPLE OF ESTIMATE



potential of node — Sum of profits of objects already taken + Sum of profits of all objects still not taken.

BRANCH & BOUND

- we explore the backtrack tree
- every node → estimate of the best soln consistent with that node

MAX (OPT ≤ estimate)

min (estimate ≤ OPT)

- prune the backtrack tree as follows;

keep track of the best integral soln found so far.

if estimate is worse than this best

⇓

prune the node

prune the node

$\boxed{\text{The better the estimate} \Rightarrow \text{smaller tree}}$

estimate → easy to compute.

$\boxed{\text{How can we get a better estimate?}}$

IP ⟶ LP relaxation

$$\boxed{OPT_{IP} \leq OPT_{LP\text{-relaxation}}}$$

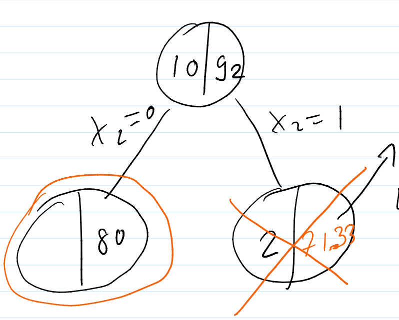
can take the relax estimate!

$\boxed{\text{How this works for knapsack}}$

$$IP \begin{cases} \max(45x_1 + 48x_2 + 35x_3) \\ 5x_1 + 8x_2 + 3x_3 \leq 10 \\ x_1, x_2, x_3 \in \{0,1\} \end{cases}$$

$$LP_{rel.} \begin{cases} x_1 = 1 \\ x_2 = 0.25 \\ x_3 = 1 \end{cases}$$

estimate $45 + 48 \cdot 0.25 + 35 = 92$



$$48 + \begin{cases} \max(45x_1 + 35x_3) \\ 5x_1 + 3x_3 \leq 2 \\ x_1, x_3 \in \{0,1\} \end{cases}$$

$\frac{45}{5} = 9$

$V \frac{35}{3} = 11.66$

$\{ x_3 = 0.66 \ (2/3)$

$$\begin{cases} x_3 = 0.66 \quad (2/3) \\ x_1 = 0 \end{cases}$$

$$48 + 35 \times \frac{2}{3}$$

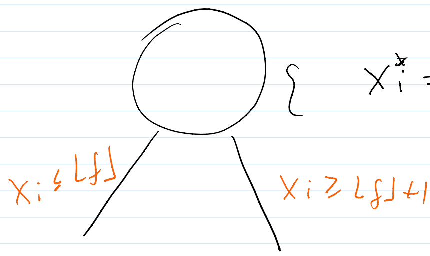
$$\parallel$$

$$48 + \frac{70}{3} = 48 + 23.33$$

$$= 71.33$$

$\boxed{\text{HEURISTIC}}$ Branch upon the "most fractional variable" in the opt. LP relaxation

$\{ x_i^* = f \}$



$\boxed{\text{When is B\&B effective}}$ When LP relaxation is strong.

$$\rho = \frac{OPT_{LP}}{OPT_{IP}}$$

as close to 1 as possible

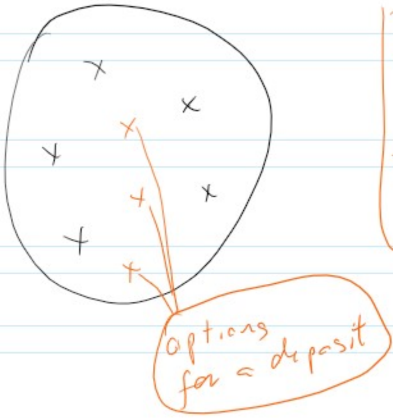
Limit case $\rho = 1$

no branching!

$OPT_{LP} = OPT_{IP}$

$\boxed{\text{Comparing two formulations for the same problem}}$

$\boxed{\text{FACILITY LOCATION}}$



c_w = cost of opening deposit w

t_{wc} = cost of supplying gas station w from deposit c

QUESTION

What deposits to open and which deposits should serve a gas station so that total cost is minimized?

$$x_w \in \{0,1\} \qquad x_w = \begin{cases} 1 & \text{open } w \\ 0 & \text{otherwise} \end{cases}$$

$$y_{w,c} \in \{0,1\} \qquad y_{w,c} = \begin{cases} 1 & c \text{ is supplied from } w \\ 0 & \text{otherwise.} \end{cases}$$

$$IP \begin{cases} \min \Big(\underbrace{\sum_w c_w x_w}_{\text{cost of opening}} + \underbrace{\sum_w \sum_c t_{wc}\, y_{wc}}_{\text{total cost of supplying.}} \Big) \\ (*) \\ \sum_w y_{w,c} = 1 \quad \{c \in C\} \\ x_w, y_{w,c} \in \{0,1\} \end{cases}$$

can only supply a gas station from an open deposit

every gas station supplied from exactly one deposit

Two ways of writing constraint (*)

Two ways of writing constraint (*)

(1) $\boxed{y_{w,c} \le x_w}$ for every (w,c)

$\Downarrow$

(2) $\boxed{\sum_c y_{w,c} \le |C|\, x_w}$ for every w

IP_1 : (*) = (1)

IP_2 : (*) = (2)

Which one of IP_1, IP_2 is better?

IP₁ is better than IP₂

Every soln 1 is a soln 2.

LP_1 , LP_2

$\rho_{IP_1} = \frac{OPT_{IP_1}}{OPT_{LP_1}}$ $\quad OPT_{IP_1} = OPT_{IP_2}$

$\vee\!/$

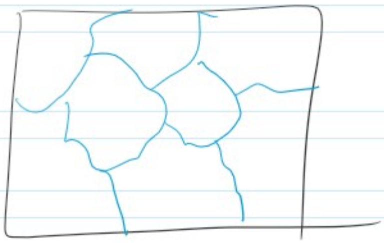
$OPT_{LP_1} \geqslant OPT_{LP_2}$

$\boxed{1 \le \rho_{IP_1} \le \rho_{IP_2}}$

better

$\boxed{\text{Tighter} \Rightarrow \text{better}}$

Example: Map Coloring



Color map with smallest # of colors

$$
\begin{cases}
\min \left(\max\limits_{c \text{ country}} \text{color}\{c\} \right) \\
\text{color}\{c_1\} \neq \text{color}\{c_2\} \quad \text{whenever } c_1 \sim c_2
\end{cases}
$$

not a linear constraint

Big-M transformation

$$X \neq Y \Leftrightarrow X \leq Y - 1 \ \underline{\text{or}} \ X \geq Y + 1$$

disjunction

How to simulate a disjunction

- use extra 0/1 variable b
- use "large integer" M

$$
\begin{cases}
X \leq Y - 1 + bM \\
X \geq Y + 1 - (1-b)M
\end{cases}
$$

Suppose $X \leq Y - 1 \Rightarrow$ choose $b = 0$

$x \geq y+1 \Rightarrow$ choose $b=1$

$$
\begin{cases}
\min(obj) \\
obj \geq color\{c\} \\
color\{c_1\} \leq color\{c_2\} - 1 + b_{c_1,c_2} M \\
color\{c_1\} \geq color\{c_2\} + 1 - (1 - b_{c_1,c_2}) M \\
obj,\ color\{c\} \geq 0 \quad \text{integral} \\
b_{c_1,c_2} \in \{0,1\}
\end{cases}
$$

PROBLEM $\quad$ LP $\rightarrow$ $b_{c_1,c_2} = 0.5$

Better Binarization

$$
b_{x,i} = \begin{cases} 1 & color\{x\} = i \\ 0 & \text{otherwise} \end{cases}
$$

$b_{x,0},\ b_{x,1},\ b_{x,2},\ b_{x,3}$

$$
\begin{cases}
\min(obj) \\
k \cdot b_{x,k} \leq obj \\
b_{x,0} + b_{x,1} + b_{x,2} + b_{x,3} = 1 & \{\forall x \in V\} \\
b_{x,i} + b_{y,i} \leq 1 & \forall i = \overline{0,3} \\
 & x \sim y
\end{cases}
$$