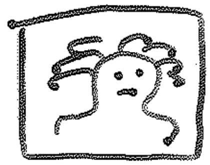
10. Branch & bound

Merchant of Venice (Shakespeare)



one portrait in one of three caskets

gold	silver	lead
?	?	?
"P is here"	"P is not here"	"P is NOT in gold casket"

$$x_i = \begin{cases} 1 & , \text{ P is in casket } i \\ 0 & , \text{ -"- NOT -"-} \end{cases}$$

$$y_i = \begin{cases} 1 & , \text{ inscription } i \text{ is true} \\ 0 & , \text{ -"- not true} \end{cases}$$

At most 1 inscription true!

(real inscriptions are @ shakespeare.mit.edu)

MOROCCO

The first, of gold, who this inscription bears,
'Who chooseth me shall gain what many men desire;'
The second, silver, which this promise carries,
'Who chooseth me shall get as much as he deserves;'
This third, dull lead, with warning all as blunt,
'Who chooseth me must give and hazard all he hath.'
How shall I know if I do choose the right?

PORTIA

The one of them contains my picture, prince:
If you choose that, then I am yours withal.

<W. Shakespeare, The Merchant of Venice>
< 1596 - 1598 >
< http://shakespeare.mit.edu/ >

B&B (c'td)

IP constraints :

$x_1 + x_2 + x_3 = 1$ (one portrait)

$y_1 + y_2 + y_3 \leq 1$ (≤ 1 true!)

$\boxed{\text{gold}}$ P is here $\Big\}$ $y_1 = x_1$

$\boxed{\text{silver}}$ P not here $\Big\}$ $y_2 = \neg x_2$ or $y_2 = 1 - x_2$

$\boxed{\text{lead}}$ P is NOT in g.c. $\Big\}$ $y_3 = \neg x_1$ or $y_3 = 1 - x_1$

$x_i, y_i \in \{0, 1\}$

- Find a feasible solution.

B&B (c'td)

Total enumeration:

(works if integer variables are constrained from above & below

$$0 \leq x_i \leq 1$$
$$0 \leq y_i \leq 1 \quad)$$

1) Enumerate all combinations of var.
2) Test feasibility
3) Return solution with best obj.

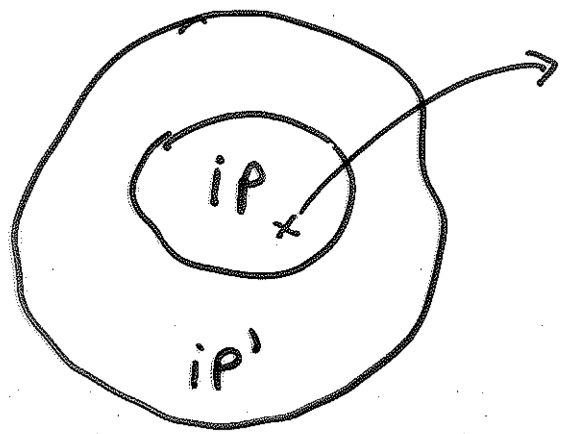
$$\underbrace{\underbrace{x_1 \quad x_2 \quad x_3}_{2^3 \text{ choices}} \qquad\qquad \underbrace{y_1 \quad y_2 \quad y_3}_{2^3 \text{ choices}}}_{2^6 \text{ choices.}}$$

B&B (c'td)

Smarter approach:

$$\left.\begin{array}{l} x_1 + x_2 + x_3 = 1 \\ x_i \in \{0,1\}. \end{array}\right\} \; ip'(x_1 \ldots x_3, y_1 \ldots y_3)$$

ip' that keeps only constraint (1) from ip is a RELAXATION of ip.



a solution feasible for ip is also feasible for ip', but not viceversa

Solve ip':

	x_1	x_2	x_3
3 candidate solutions	1	0	0
	0	1	0
	0	0	1

B&B (c'ed)

Solve ip

- for each $\{x_1, x_2, x_3\}$ candidate ip',
  find $y_1 \dots y_3$ to satisfy ip.

$\Downarrow$

3 easier ip subproblems

a) $x^T = (1\ 0\ 0)$

$$\Rightarrow \left.\begin{aligned} y_1 &= x_1 = 1 \\ y_2 &= 1 - x_2 = 1 \\ y_3 &= 1 - x_1 = 0 \end{aligned}\right\} \quad y_1 + y_2 + y_3 = 2 \not\leq 1$$

Problem a) infeasible

b) $x^T = (0\ 1\ 0)$

$$\Rightarrow \left.\begin{aligned} y_1 &= x_1 = 0 \\ y_2 &= 1 - x_2 = 0 \\ y_3 &= 1 - x_1 = 1 \end{aligned}\right\} \quad y_1 + y_2 + y_3 \leq 1$$

Problem b) feasible, DONE.

B&B (c'ed)

A similar but general approach (B&B):

Let IP:

$$\min c^T x$$
$$Ax = b$$
$$x \in \mathbb{Z}_+$$

- start by solving a relaxation of IP:

$$(P): \begin{cases} \min c^T x \\ Ax = b \\ x \geq 0 \end{cases}$$ (LP relaxation)

(other relaxations will do as well)

- let $x^{*T} = (\theta_1\ \theta_2 \ldots \theta_n)$ be the optimal solution of P.

→ if $\theta_i \in \mathbb{Z}_+$ for all $i \in \{1, \ldots n\}$ we are DONE.

→ suppose θ_i is fractional.

B&B (c'ed)

⋮

P : a relaxation of IP. Let's call it the CURRENT Pb.

$(\theta_1 \dots \theta_n)$: optimal (LP) solution of P

$z(P)$: optimal cost of P.

θ_i : fractional

→ define two new "subproblems".

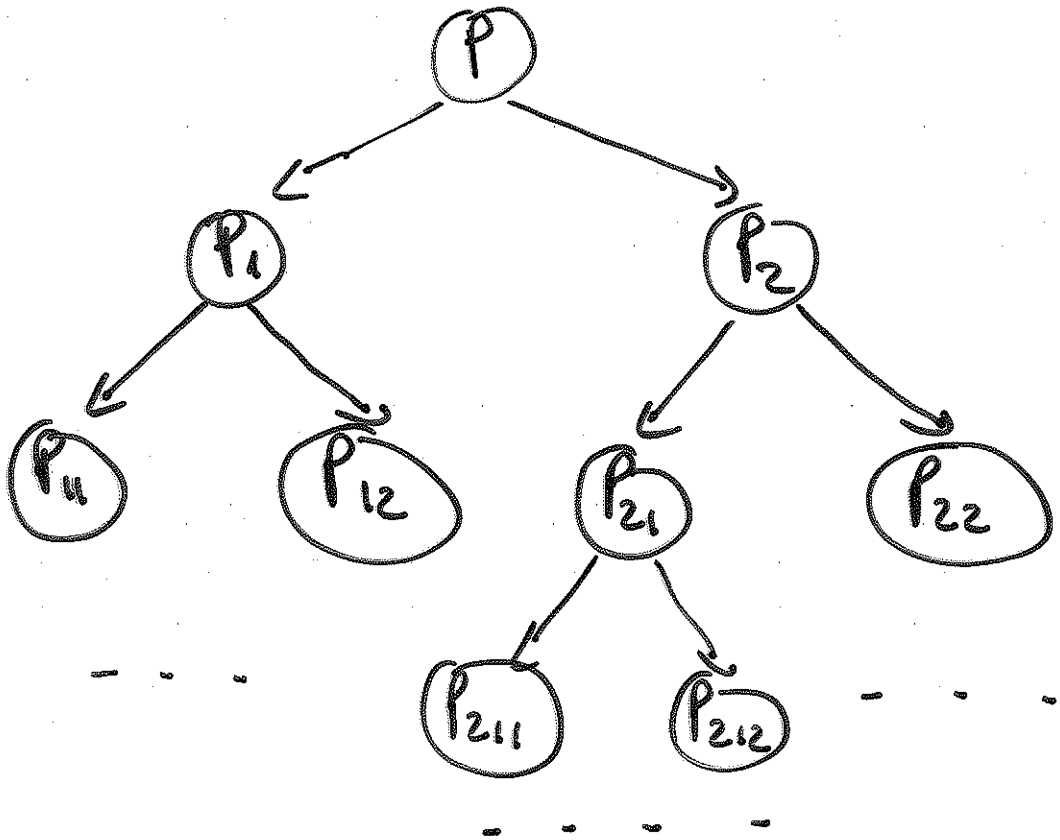
$P_1 : P \cup \{ \xi_i \leq \lfloor \theta_i \rfloor \}$ $\qquad$ $P_2 : P \cup \{ \xi_i \geq \lceil \theta_i \rceil \}$

(this is called branching)

- recursively "solve" P_1 & P_2.

B&B (c'ed)

We obtain a B&B tree :

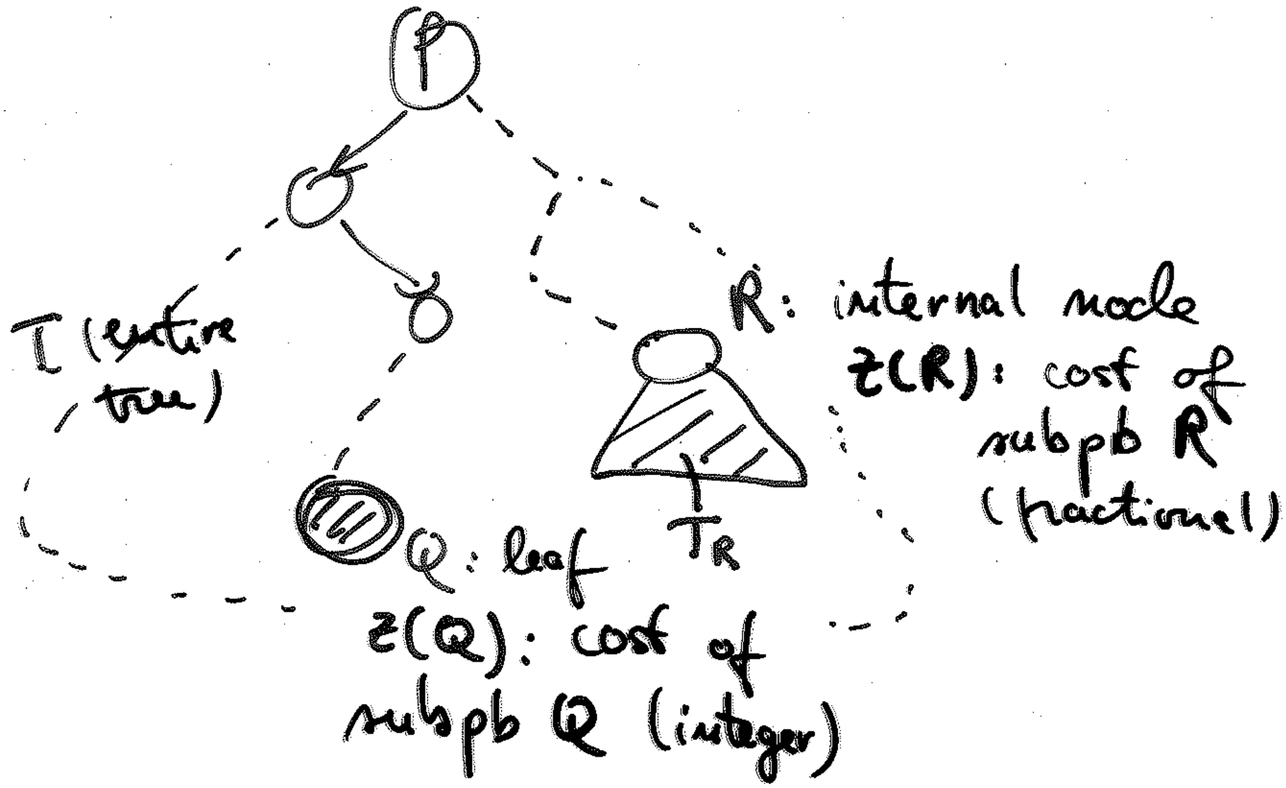


leaves contain subproblems wose LP solutions turn out integer.

IP solution = solution of leaf subproblem with smallest cost.

B&B (c'ed)

Efficient implementation



OBS 1 $z(Q)$ = upper bound on the optimal IP.

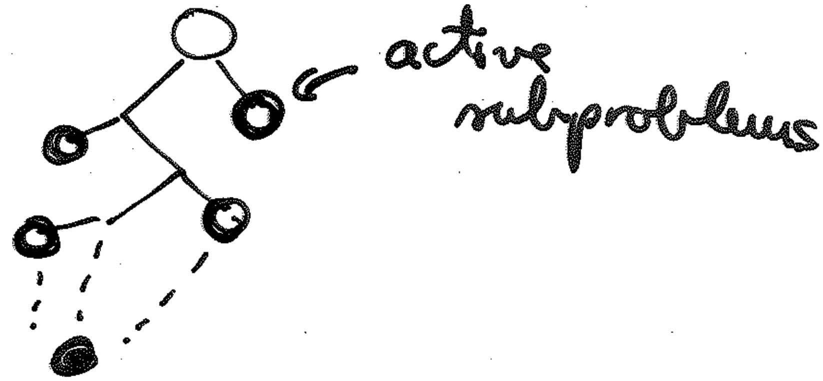
OBS 2 $z(R)$ = lower bound on the cost of any leaf subproblem in T_R

OBS 3 if $z(R) \geq z(Q)$ we can prune T_R since T_R cannot contain a better solution than Q.

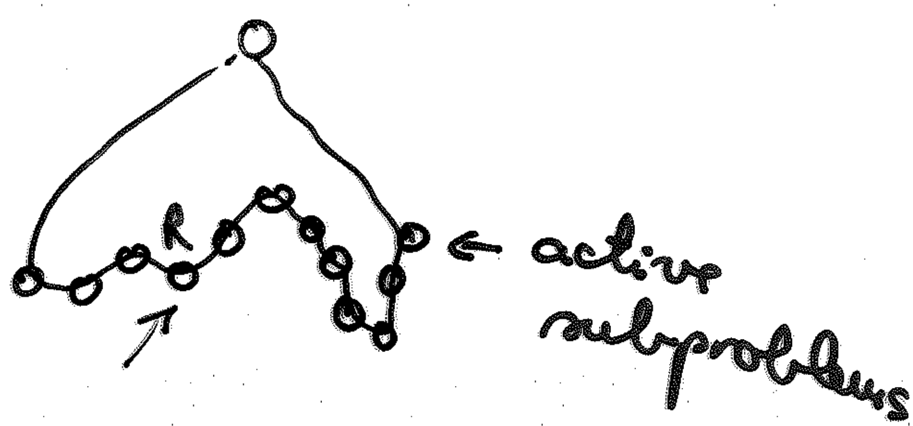
B&B (c'ed)

Search strategies:

- depth first



- priority



– next pb selected for exploration from the set of active pb has min. cost
(this way we hope to obtain a good upper bound that will prune many nodes)