Branch and Bound

(Wolsey : Chapter 7 (7.1, 7.2))

Consider

$$(P) \qquad z = \max_{x \in S} c(x)$$

How can we break (P) into a series of smaller (and easier) problems, solve the smaller problems, and then put the information together to solve (P) ?

Divide and conquer approach

Proposition : Let $S = S_1 \cup S_2 \cup \dots \cup S_K$ be a decomposition of S, and let $z^k = \max\{c(x) : x \in S_k\}$, $k = 1, \dots K$.

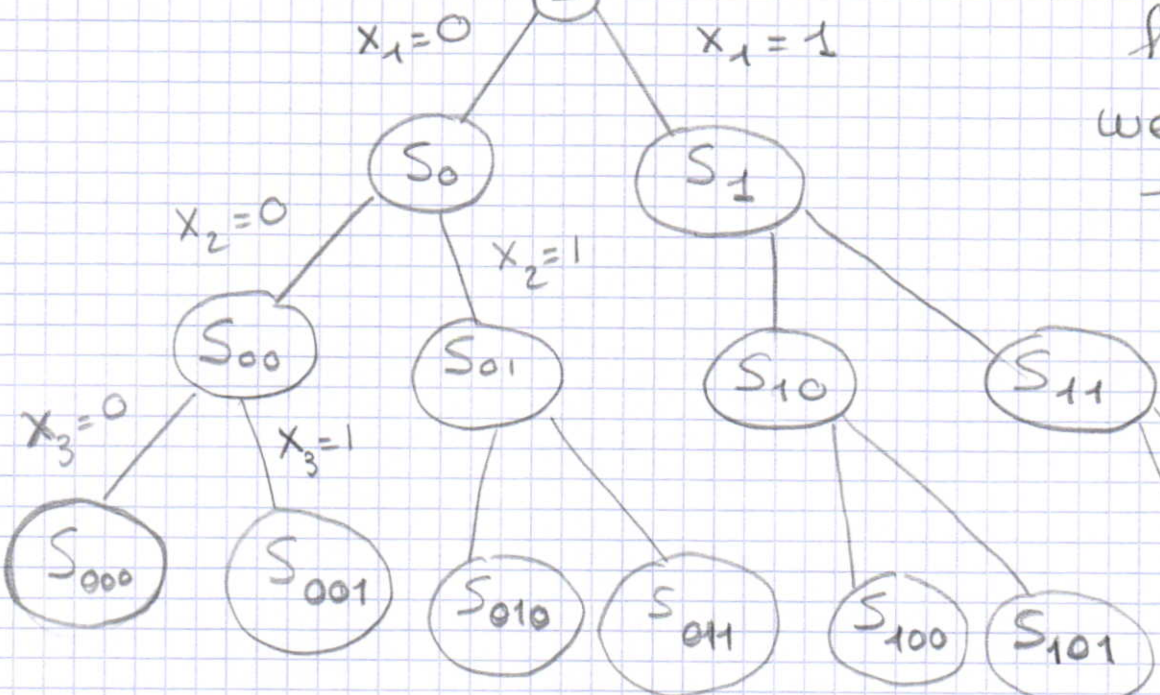
Then $z = \max\limits_k z^k$.

- A typical way is to decompose S via an enumeration tree
- However: a complete enumeration is usually impossible (we can not divide indefinitely)

<u>So</u>:

- how can we use some bounds on $\{z^k\}$ intelligently?
- how can we put together bound information?

<u>example</u>: binary enumeration tree for $S \subseteq \{0,1\}^3$



how can we "prune" this tree?

Implicit enumeration

Proposition : Let $S = S_1 \cup .. \cup S_K$ be a decomposition of S, and let $z^k = \max\{c(x) : x \in S_k\}$, $\bar{z}^k$ be an upper bound on z^k, and $\underline{z}^k$ be a lower bound on z^k, $k = 1, \ldots K$.

Then :

$$\boxed{\bar{z} = \max_k \bar{z}^k}$$ is an upper bound on z

and

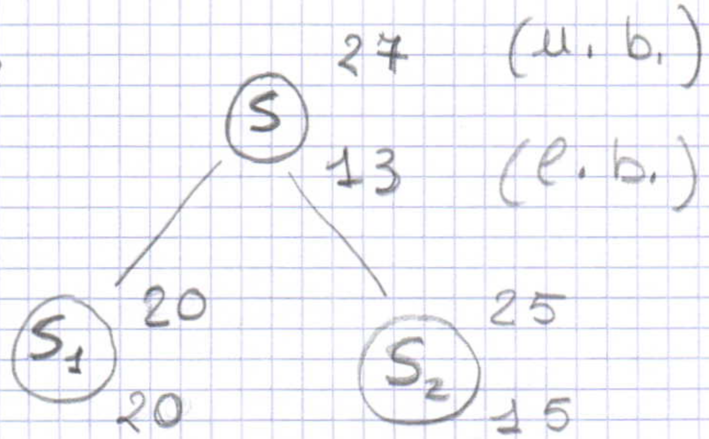
$$\boxed{\underline{z} = \max_k \underline{z}^k}$$ is a lower bound on z

- So, bound information (partial information) about subproblems can be put together to derive bounds on z !

- What can be deduced from these bounds, and which sets need further examination to compute z ?

Examples

(1) max



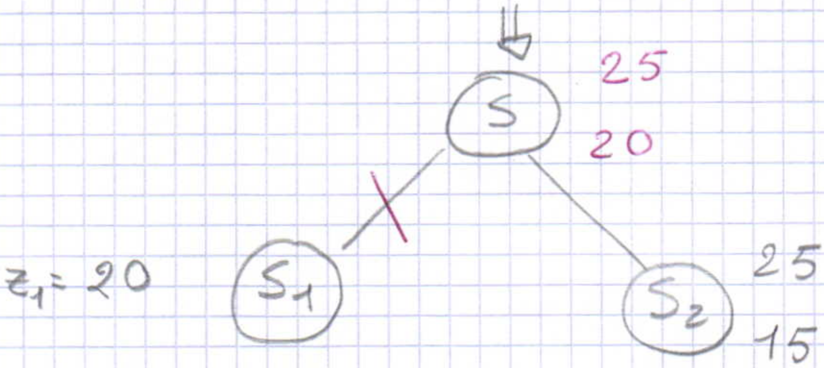
Based on the previous proposition:

$$\bar{z} = \max_k \bar{z}^k = \max\{20, 25\} = 25$$

$$\underline{z} = \max_k \underline{z}^k = \max\{20, 15\} = 20$$

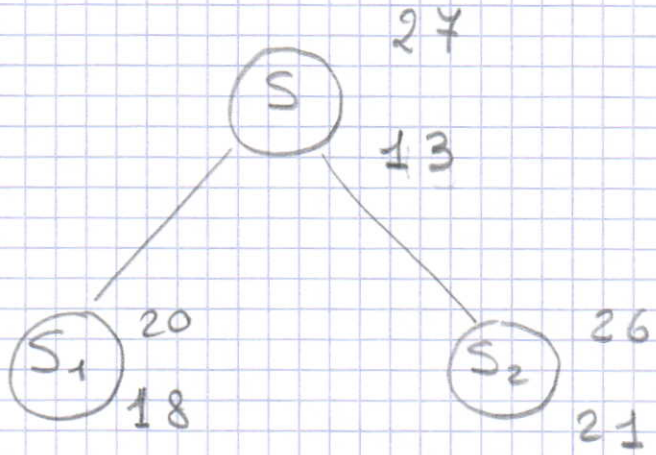
Further observe that, since the upper and lower bounds on z_1 are equal, then $z_1 = 20$, and so there is no further reason to examine S_1: the branch S_1 of the enumeration tree can be <u>pruned by optimality</u>

$\Downarrow$



Pruning by optimality

2 max

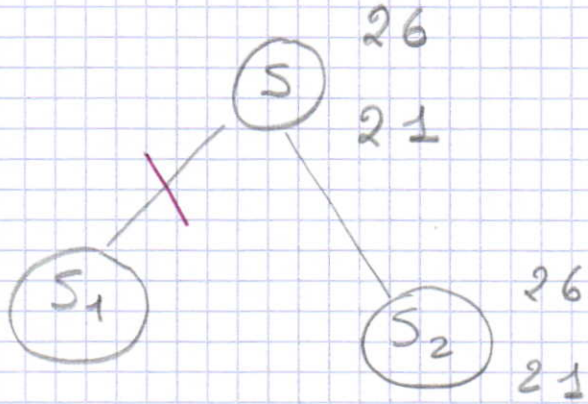


$$\overline{z} = \max\{20, 26\} = 26$$

$$\underline{z} = \max\{18, 21\} = 21$$

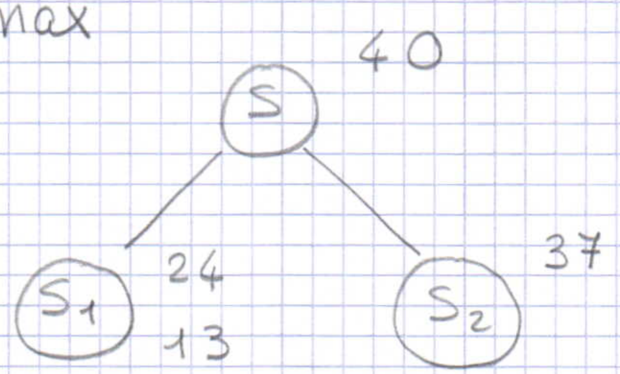
Further observe that, since $z \geq 21$, and the upper bound $\overline{z}_1 = 20$, then no optimal solution of (P) can belong to S_1: the branch S_1 of the enumeration tree can be <u>pruned by bound</u>

$\Downarrow$



Pruning by bound

3 max

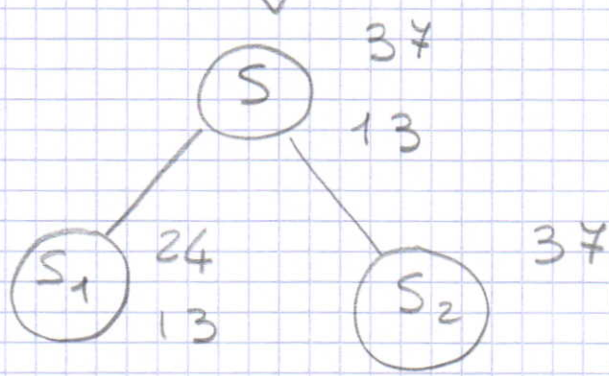


$\bar{z} = \max\{24, 37\} = 37$

$\underline{z} = \max\{13, -\infty\} = 13$

Here no further conclusion can be drawn: we need to explore both S_1 and S_2

$\Downarrow$



So, we can list at least 3 reasons that allow us to "prune" the tree, and thus enumerate many solutions implicitely:

1) <u>pruning by optimality</u>:

$$z_t = \max_{x \in S_t} c(x) \qquad \text{for some } t$$

has been solved;

2) <u>pruning by bound</u>:

$$\bar{z}_t \leq \underline{z} \qquad \text{for some } t$$

3) <u>pruning by infeasibility</u>:

$$S_t = \emptyset \qquad \text{for some } t.$$

Based on the above ideas, we can design an implicit enumeration framework, or Branch and Bound.

Branch and Bound framework

Procedure B & B (P, z)

begin

$Q := \{(P)\}; \quad z := -\infty;$

* Q is the set of active ($\equiv$ no pruned) nodes of the tree *

repeat

$(P') := Next(Q); \quad Q := Q \setminus \{(P')\};$

$\bar{z} := RELAX(P');$

if $\bar{z} > z$ **then**

  begin

  $\underline{z} := HEURISTIC(P');$

  if $\underline{z} > z$ **then** $z := \underline{z};$

  if $\bar{z} > \underline{z}$ **then**

    $Q := Q \cup BRANCH(P')$

  end

until $Q = \emptyset$

Questions to be addressed to have a well-defined Branch and Bound algorithm

(1) <u>RELAX</u> : upper (dual) bound

< via a relaxation approach >

<u>HEURISTIC</u> : lower (primal) bound

< via a heuristic approach >

- strong (upper) bounds
- efficient procedures (quite rapid)
- RELAX and HEURISTIC possibly interconnected
- reoptimization techniques if possible

(2) <u>NEXT</u> : order to examine subproblems

(= visiting strategies of the enumeration tree)

- <u>topological visits</u> :
  - breadth-first (Q is a queue)
  - depth-first (Q is a stack)

- information based visits:

  e.g. best - first ("promising" node)

(3) BRANCH: how should S be decomposed?

- completeness is necessary:

$$S = S_1 \cup S_2 \cup \ldots \cup S_K$$

- additional (not necessary) properties:

- partitioning:

$$S_i \cap S_j = \emptyset \qquad \forall i \neq j$$

* a unique path in the enumeration tree for each (examined) solution *

- balancing:

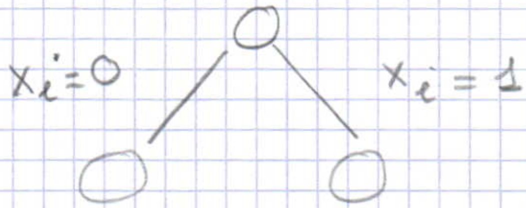
$$|S_i| \simeq |S_j| \qquad \forall i \neq j$$

* balanced enumeration tree *

* relevant if a depth-first visit is adopted *

- compatibility with RELAX and HEURISTIC:

example: Knapsack



the two subproblems are still (reduced) Knapsack problems: we can use the same bounding procedure at each node!

- few sons per node: too many sons may require a large amount of memory

+

Preprocessing: to accelerate the execution

example: Knapsack

- if $a_i > b$ we can fix $x_i = 0$
- if $c_i \leq 0$ and $a_i \geq 0$ " " $x_i = 0$

⋮

Examples

1 Binary Knapsack

$$(P) \quad \max \; 8x_1 + 5x_2 + 5x_3 + 3x_4 + x_5$$
$$4x_1 + 3x_2 + 4x_3 + 3x_4 + 2x_5 \leq 12$$
$$x_i \in \{0,1\} \quad i = 1, \dots 5$$

RELAX : linear relaxation

HEURISTIC : greedy heuristic (CUD)

BRANCH : on the only fractional variable of the LP optimal solution

NEXT : breadth-first, by giving priority to the subproblem obtained by fixing the fractional variable to 1

Notation (at each node):

- x^* : optimal solution of the LP
- $\bar{x}$: solution of greedy heuristic
- $\bar{z}$: upper bound ($\bar{z} = cx^*$)
- $\underline{z}$: lower bound ($\underline{z} = c\bar{x}$)
- z : current best solution value

Note that

$$\frac{c_1}{a_1} \geq \frac{c_2}{a_2} \geq \frac{c_3}{a_3} \geq \frac{c_4}{a_4} \geq \frac{c_5}{a_5}$$

<u>Initialization</u> : $Q := \{(P)\}$; $\underline{z} := -\infty$

$\boxed{\text{Radix node}}$ P

$x^* = (1, 1, 1, \frac{1}{3}, 0)$ $\quad \bar{z} = 19$

$\bar{x} = (1, 1, 1, 0, 0)$ $\quad \underline{z} = 18$

Since $\underline{z} > \underline{z}(= -\infty)$ then $\underline{z} = 18$

Since $\bar{z} > \underline{z}$, then BRANCH on x_4

$\boxed{x_4 = 1}$ P_1

$x^* = (1, 1, \frac{1}{2}, 1, 0)$ $\quad \bar{z} = 18 + \frac{1}{2}$

$\bar{x} = (1, 1, 0, 1, 1)$ $\quad \underline{z} = 17$

Since $\bar{z} > \underline{z}$, then BRANCH on x_3

$\boxed{x_4 = 0}$ P_2

$x^* = (1, 1, 1, 0, \frac{1}{2})$ $\quad \bar{z} = 18 + \frac{1}{2}$

$\bar{x} = (1, 1, 1, 0, 0)$ $\quad \underline{z} = 18$

Since $\bar{z} > \underline{z}$, then BRANCH on x_5

$\boxed{x_4 = 1, x_3 = 1}$ P_3

$x^* = (1, \frac{1}{3}, 1, 1, 0)$ $\bar{z} = 17 + \frac{2}{3}$

Since $\bar{z} \leq z = 18$, then pruning by bound

$\boxed{x_4 = 1, x_3 = 0}$ P_4

$x^* = (1, 1, 0, 1, 1)$ $\bar{z} = 17$

pruning by optimality (and also by bound)

$\boxed{x_4 = 0, x_5 = 1}$ P_5

$x^* = (1, 1, \frac{3}{4}, 0, 1)$ $\bar{z} = 17 + \frac{3}{4}$

Since $\bar{z} \leq z = 18$, then pruning by bound

$\boxed{x_4 = 0, x_5 = 0}$ P_6

$x^* = (1, 1, 1, 0, 0)$ $\bar{z} = 18$

pruning by optimality (also by bound)

$Q = \emptyset$ STOP

$x = (1, 1, 1, 0, 0)$, corresponding to $z = 18$, is an optimal solution

Properties :

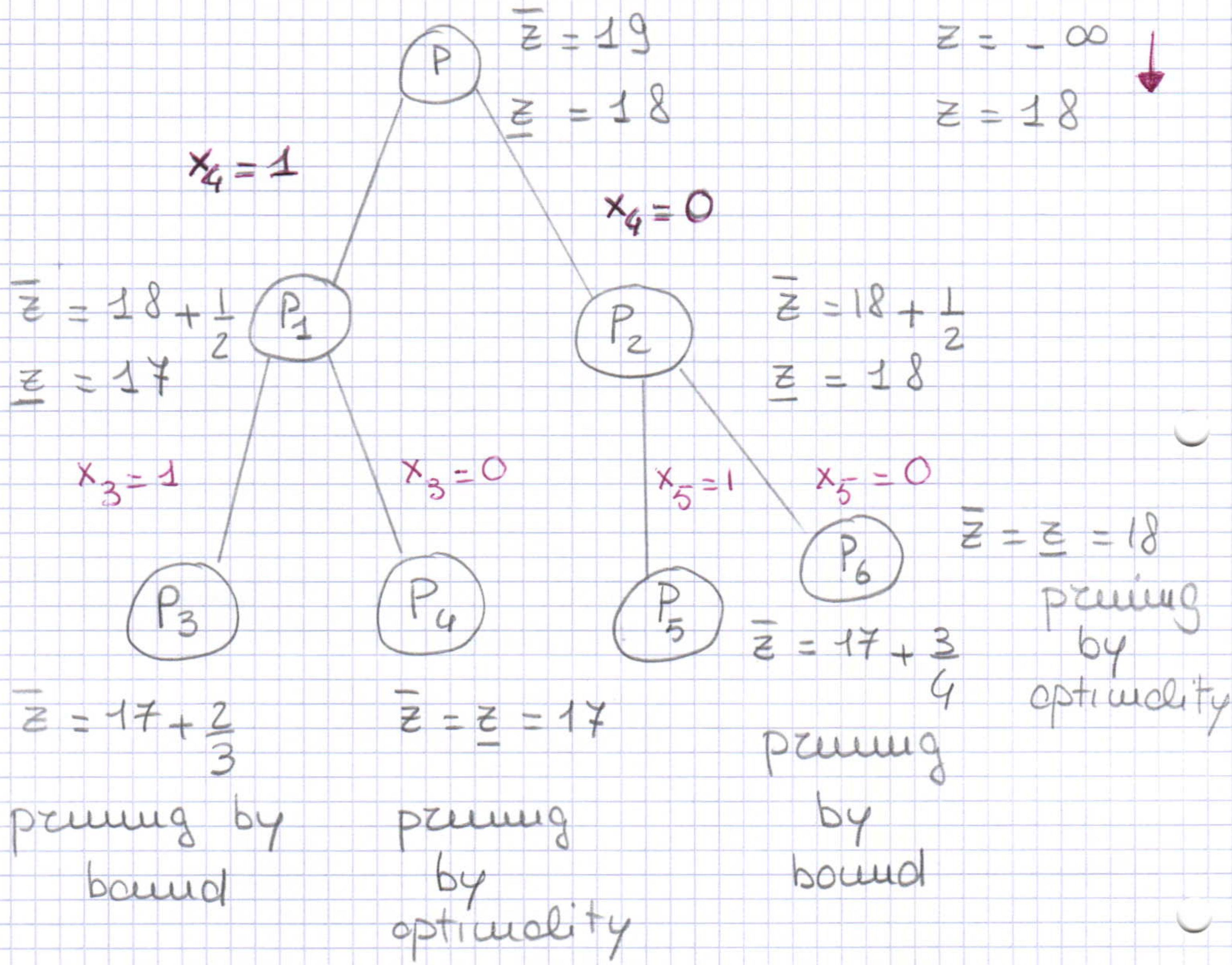
- BRANCH : partitioning rule
- Connection between RELAX and HEURISTIC (both based on the CUD order)
- BRANCH is compatible with RELAX and HEURISTIC : each subproblem is a (reduced) Knapsack problem
- The upper bound may strictly decrease along each path (the optimal LP solution is cut off from the two subproblems)

Note that

$$\max\{\bar{z}(P_1), \bar{z}(P_2)\} \leq z(P) \leq \max\{\underline{z}(P_1), \underline{z}(P_2)\}$$

decreases increases

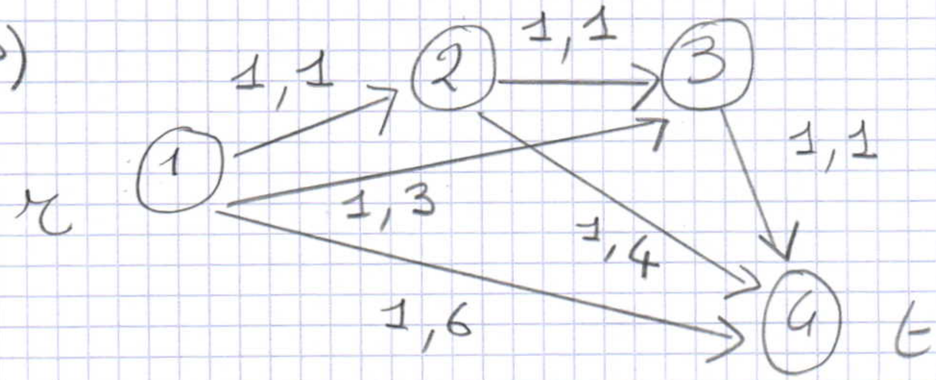
The corresponding enumeration tree

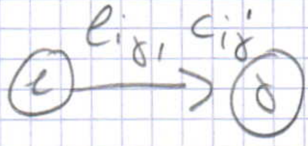


Would it be possible to prune P_1 and P_2, so generating only two levels of the tree?

2 Constrained shortest path

(P)





$L = 2$

RELAX : shortest path computation

HEURISTIC : —

BRANCH : partitioning based on the shortest path

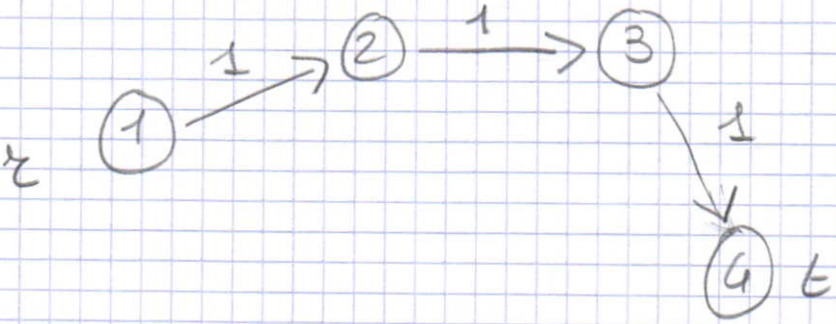
NEXT : breadth-first

<u>Initialization</u> : $Q := \{(P)\}$; $z := +\infty$

<u>Observation</u> : <u>minimization problem</u>

Radix node   P

shortest path from r to t



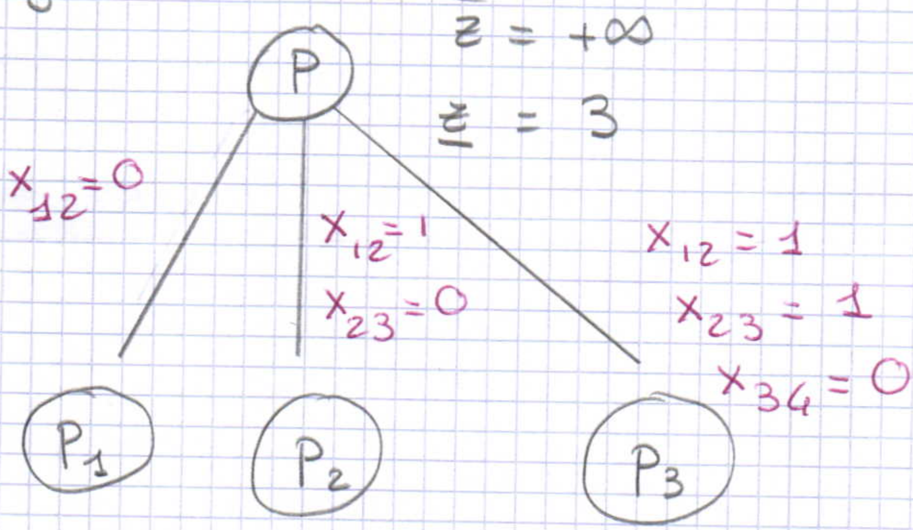
P_{rt}

$\underline{z} = 3$ lower bound

$\overline{z} = +\infty$ no heuristic

$\underline{z} < \overline{z}$ then BRANCH

Since $\sum_{(i,j)\in P_{rt}} l_{ij} = 3 > 2$, P_{rt} is not feasible, so define the following branching rule:



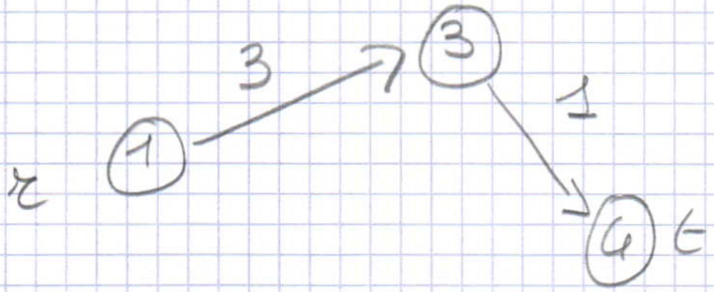
$z = +\infty$

current enumeration tree

P_1

P_1 is a constrained shortest path in a reduced graph (delete (1,2) from the graph)

shortest path from r to t



$\underline{z} = 4$

Since $\sum_{(i,j)\in P_{rt}} l_{ij} = 2 = L$, then P_{rt} is a feasible path:

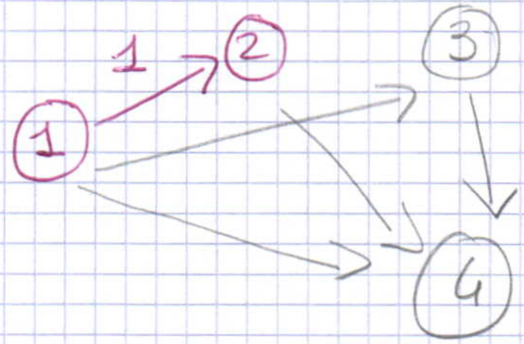
$\bar{z} = 4$

Since $\bar{z} < z \, (= +\infty)$, then $z = 4$

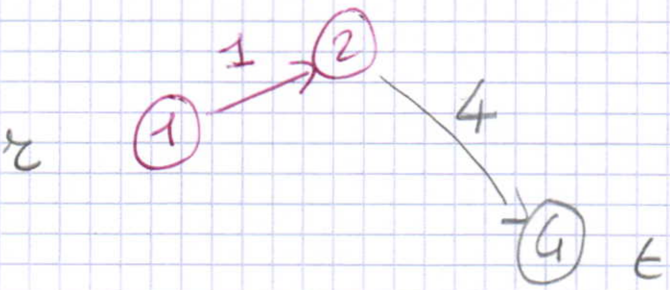
Since $\underline{z} = \bar{z}$, then pruning by optimality

$\boxed{P_2}$

P_2 ? delete (2, 3) and "imposes" (1, 2):



Equivalently: delete (2, 3), (1, 2) and the arcs incident node 1, and look for a shortest path from 2 to 4:



P_{rt}

$\underline{z} = 1 + 4 = 5$

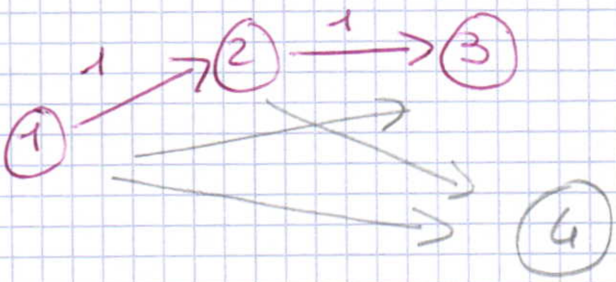
Since $\sum_{(i,j) \in P_{rt}} l_{ij} = 2 = L$, then P_{rt} is a feasible path:

$z = 4 < 5$ (no improvement of z)

Since $\underset{5}{\underset{\shortparallel}{\underline{z}}} > \underset{4}{\underset{\shortparallel}{z}}$ then pruning by bound (also by optimality)

P_3

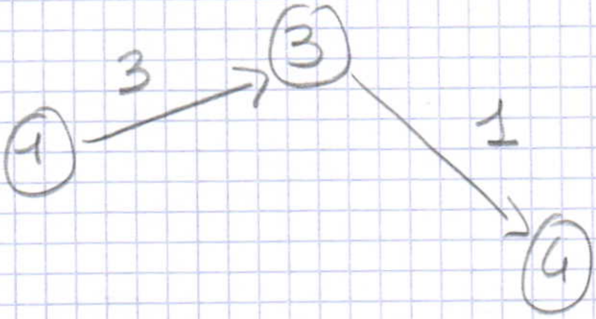
"imposes" the subpath (1, 2, 3) and delete (3, 4):



Equivalently: look for a shortest path from 3 to 4 by removing nodes 1 and 2 and arc (3, 4): no path

<u>pruning by infeasibility</u>

$Q = \emptyset$: $z = 4$

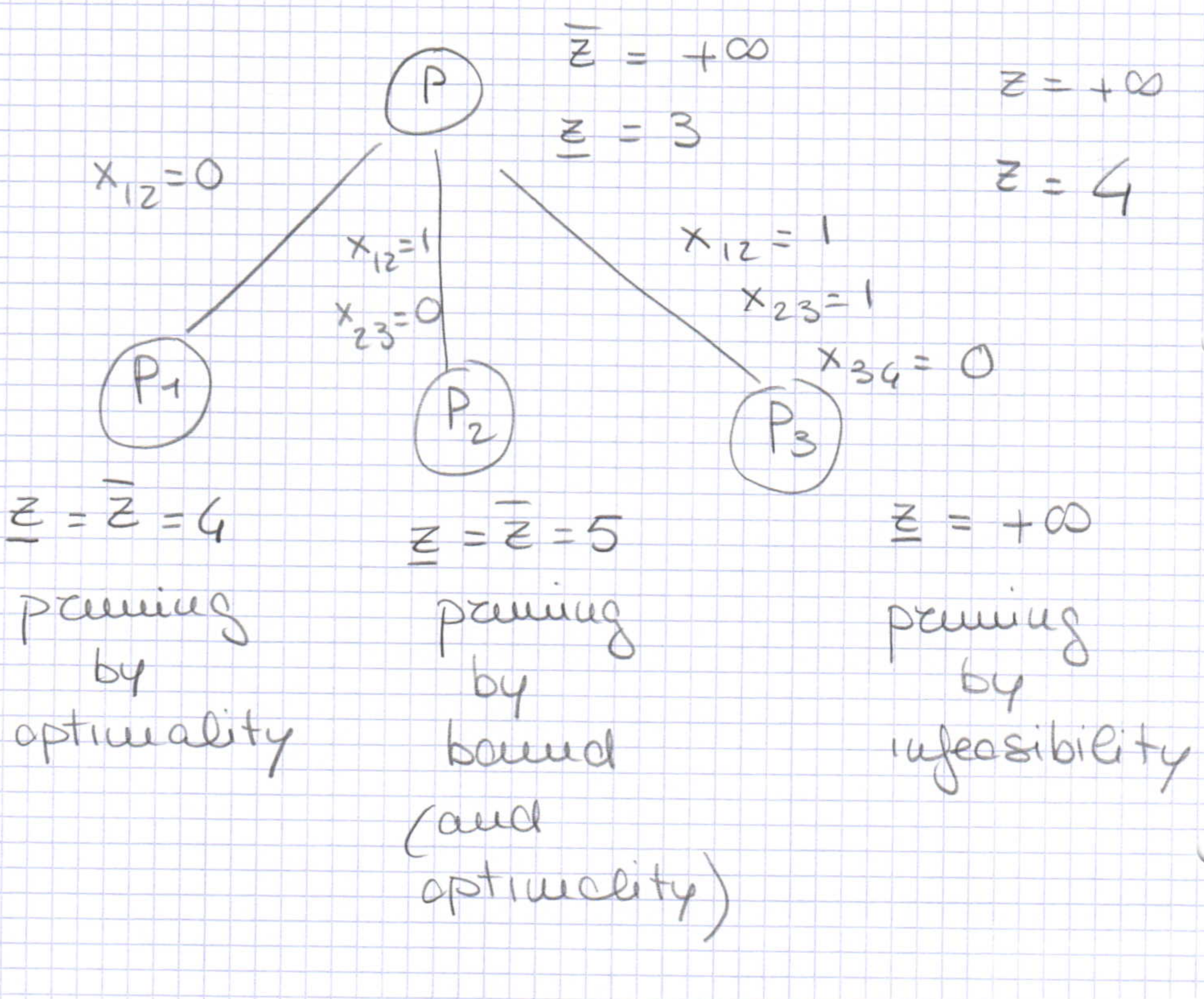


constrained shortest path

Properties :

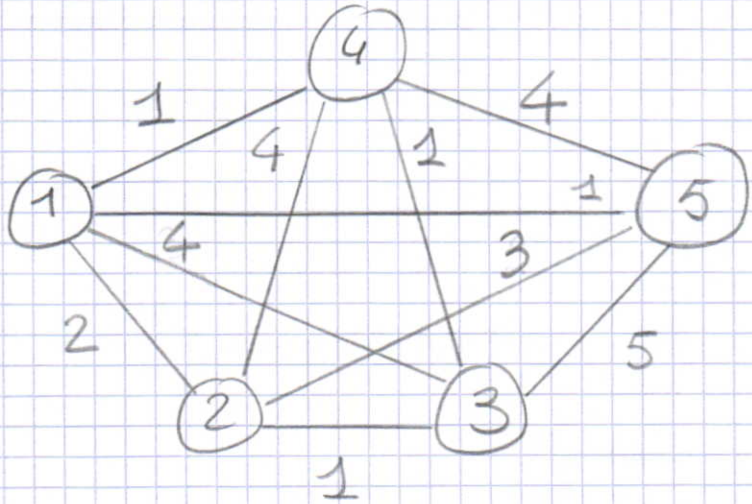
- BRANCH : partitioning rule
- BRANCH is compatible with RELAX : each subproblem is a (reduced) constrained shortest path
- The lower bound may strictly increase along each path of the tree

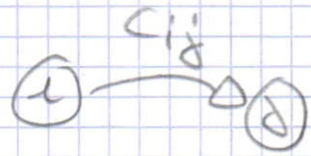
The corresponding enumeration tree



3 TSP

(P)





RELAX :

ILP formulation on $G=(N,A)$:

$$\min \sum_{(i,j)\in A} c_{ij}\, x_{ij}$$

$$\sum_{j:(i,j)\in A} x_{ij} = 2 \quad \forall i \in N \qquad (\circ)$$

connection &

<no subtour constraints> e.g. cut constraints

$$x_{ij} \in \{0,1\} \quad \forall (i,j)\in A$$

where:

$$x_{ij} = \begin{cases} 1 & \text{if } (i,j) \text{ belongs to the tour} \\ 0 & \text{otherwise} \end{cases} \qquad \forall (i,j)\in A$$

If we relax (= remove) constraints (•), then we get a Minimum Spanning Tree (MST): greedy algorithms (e.g. Kruskal)

HEURISTIC : no heuristic

BRANCH : see next

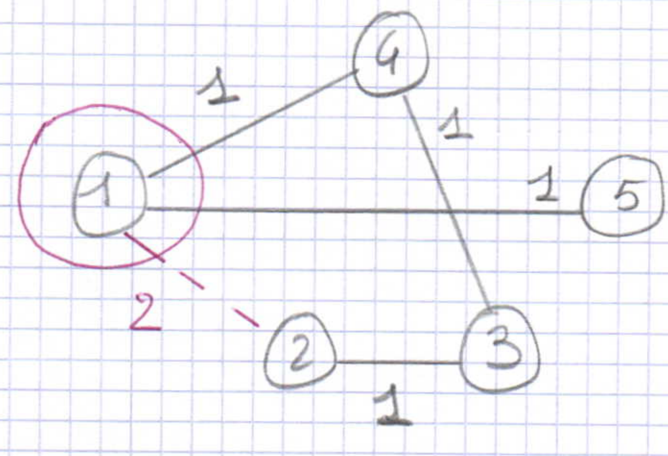
NEXT : breadth-first visit

Initialization : $Q := \{(P)\}$; $z := +\infty$

minimization

P

MST

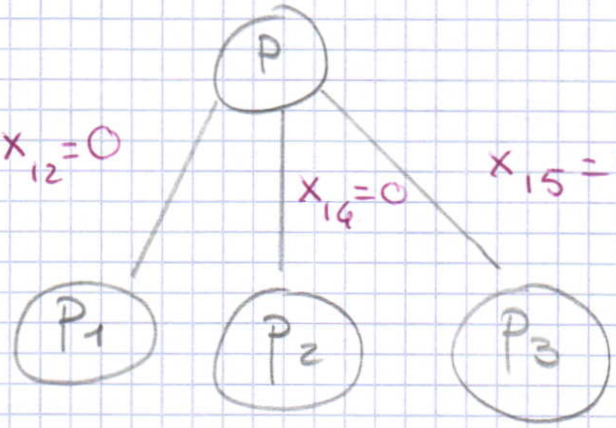


$\underline{z} = 6$

We can refine the bound by adding the minimum cost arc not belonging to the MST: MS1T (Minimum Spanning 1-Tree)

$\bar{z} = +\infty$

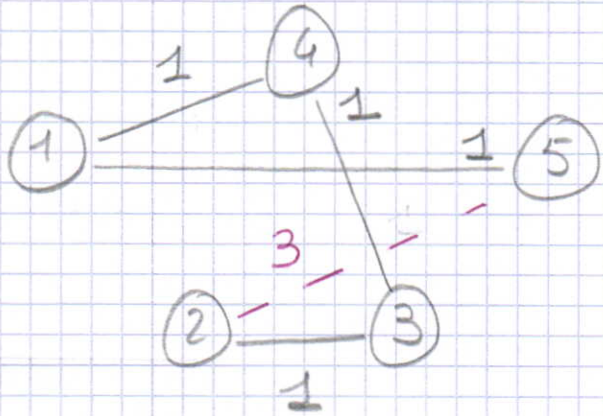
Since $\underline{z} < \bar{z}$ then BRANCH:



it is not a partitioning!

$\boxed{P_1}$

MS1T on a reduced graph (remove (1,2))



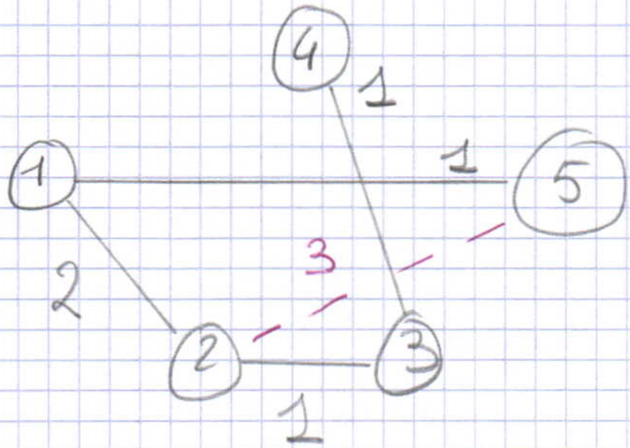
$\underline{z} = 7$

This is a Hamiltonian cycle (so a feasible solution): $z = 7$

Since $\bar{z} = \underline{z} = 7$, then <u>pruning by optimality</u>

$\boxed{P_2}$

MS1T

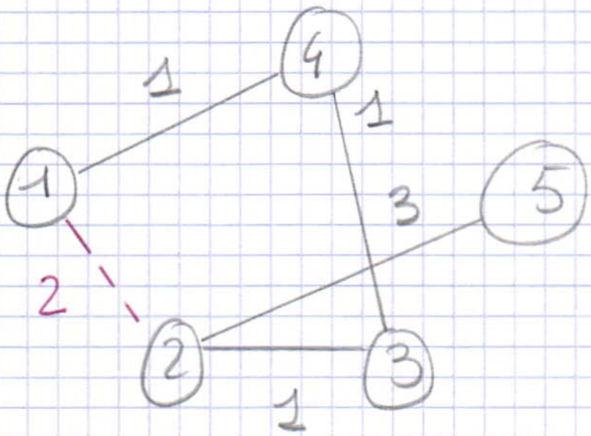


$\underline{z} = 8$

Since $\underline{z} = 8 > z = 7$, then <u>pruning by bound</u>

$\boxed{P_3}$

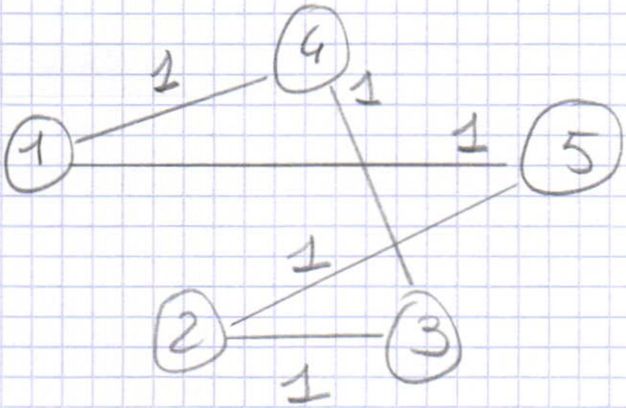
MS1T



$\underline{z} = 8$

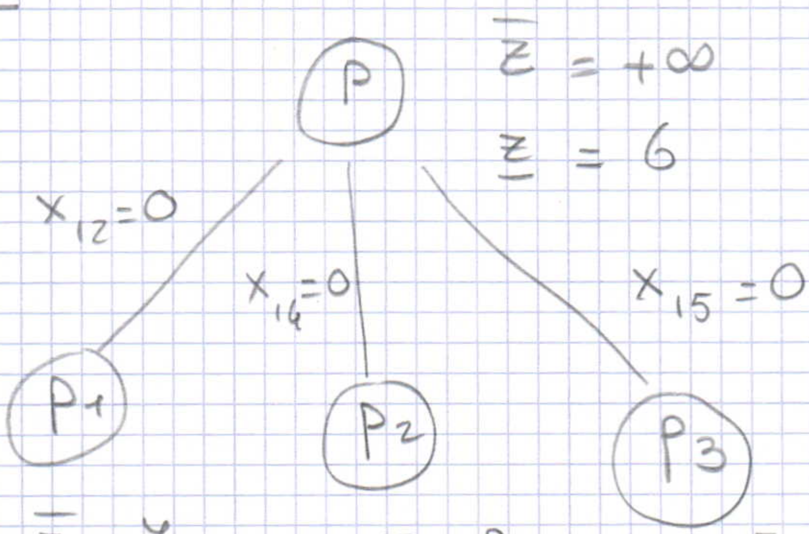
Since $\underline{z} = 8 > z = 7$, then <u>pruning by bound</u>

$\Phi = \emptyset$: $z = 7$



optimal solution

The corresponding enumeration tree



$z = +\infty$

$z = 7$

P_1	P_2	P_3
$\underline{z} = \bar{z} = 7$	$\underline{z} = 8$	$\underline{z} = 8$
pruning by optimality	pruning by bound	pruning by bound