4.1 Branch-and-bound method

we step back from the linear programming problem

Consider a generic optimization problem

$$\min\{c(x) : x \in X\}$$

Idea:

Instead of solving the problem as a whole

Reduce the solution of a difficult problem to that of a sequence of simpler subproblems by (recursive) partition of the feasible region X.

- Applicable to discrete and continuous optimization problems.
- Two main components: branching and bounding.

4.1 Branch-and-bound method

- **Branching:**
  Partition X into k subsets

$$X = X_1 \cup \ldots \cup X_k \quad (\text{with } X_i \cap X_j = \emptyset \text{ for each pair } i \neq j)$$

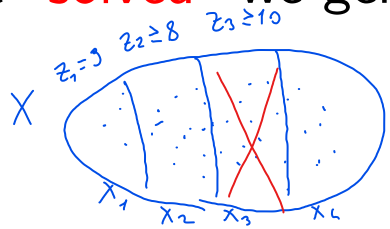
  and let

$$z_i = \min\{c(x) : x \in X_i\}, \text{ for } i = 1, \ldots, k.$$

  Clearly $z = \min\{c(x) : x \in X\} = \min\{z_1, \ldots, z_k\}$

- **Bounding:**
  For each subproblem $z_i = \min\{c(x) : x \in X_i\}$:
  - Determine an optimal solution of $\min\{c(x) : x \in X_i\}$ (explicit), or --> we are lucky and we found the optimal solution
  - Prove that $X_i = \emptyset$ (explicit), or
  - Prove that $z_i \geq z'$ = objective function value of the best feasible solution found so far (implicit).

  If the subproblem is not "solved" we generate new subproblems by further partitioning.



4.1 Branch-and-bound method

4.1.1 Branch-and-bound for ILP

Given an ILP $\min\{c^T x : Ax = b, x \geq 0 \text{ integer}\}$.

Branching:

Partition X into subregions (subdivision in exhaustive and exclusive subregions).
Let $\bar{x}$ denote an optimal solution for the linear relaxation of the ILP

then, either the optimal solution is fully integer --> no need to solve ILP
or at least one is non integer

$$\min\{c^T x : Ax = b, x \geq 0\}$$

and $z_{LP} = c^T \bar{x}$ denote the corresponding optimal value.
If $\bar{x}$ is integer, $\bar{x}$ is also optimal for ILP. Otherwise, $\exists \bar{x}_h$ fractional and we consider 2 subproblems:

$$ILP_1: \min\{c^T x : Ax = b, x_h \leq \lfloor \bar{x}_h \rfloor, \; x \geq 0 \text{ integer}\}$$

$$ILP_2: \min\{c^T x : Ax = b, x_h \geq \lfloor \bar{x}_h \rfloor + 1, \; x \geq 0 \text{ integer}\}$$

Bounding:

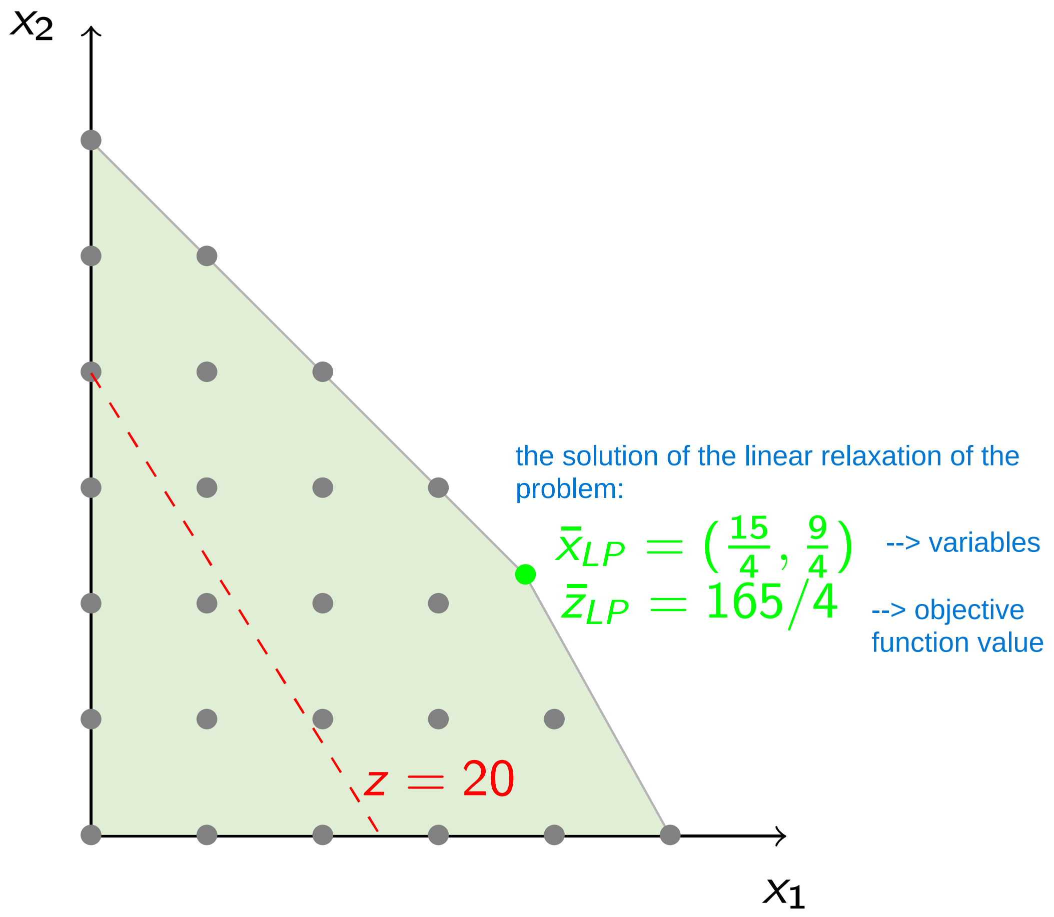
Determine a lower "bound" (if minimization ILP) on the optimal value z_i of a subproblem of ILP by solving its linear relaxation.

4.1 Branch-and-bound method

Example

$$\begin{array}{ll} \max & z_{ILP} = 8x_1 + 5x_2 \\ \text{s. t.} & x_1 + x_2 \leq 6 \\ & 9x_1 + 5x_2 \leq 45 \\ & x_1, x_2 \geq 0, \text{ integer} \end{array}$$

Clearly $z_{LP} \geq z_{ILP}$



Since $\bar{x}_1$ and $\bar{x}_2$ are fractional, select one of them for branching. For instance, x_1.

4.1 Branch-and-bound method

Example (cont.)

The feasible region X is partitioned into X_1 and X_2 by imposing:

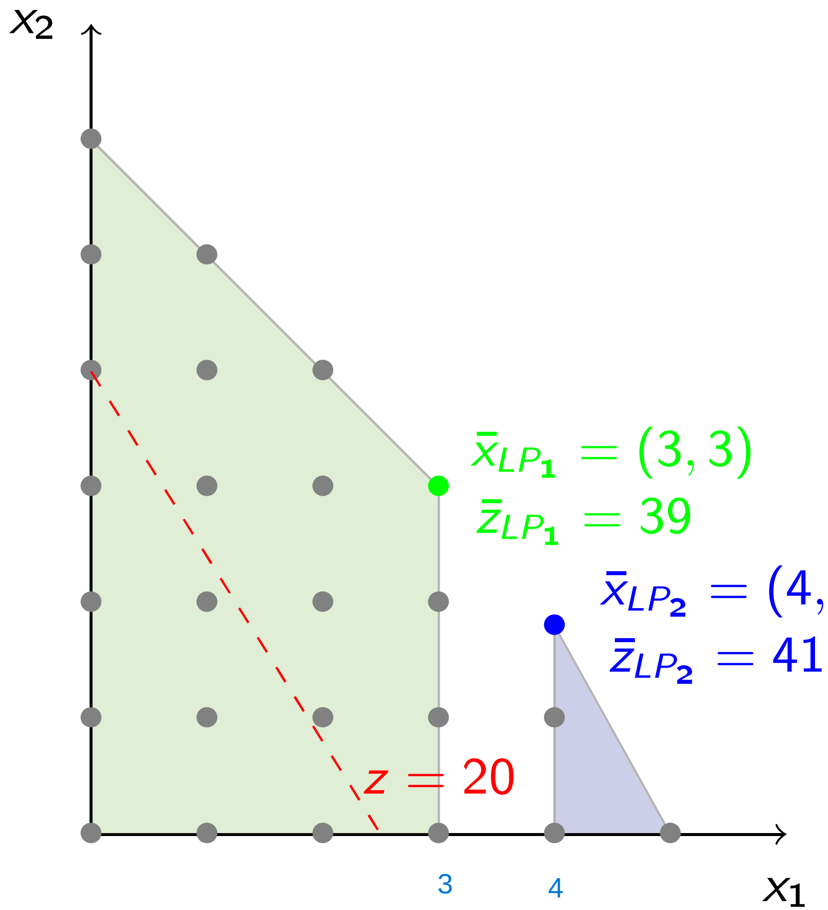
$$x_1 \leq \lfloor \overset{15/4}{\bar{x}_1} \rfloor = 3 \text{ or } x_1 \geq \lfloor \bar{x}_1 \rfloor + 1 = 4 \text{ (exhaustive and exclusive constraints)}$$

Subproblem S_1: subregion X_1

Subproblem S_2: subregion X_2

$\bar{x}_{LP_1}$ is integer $\Rightarrow$ Integer solution!

$\Rightarrow z_{ILP_1} = z_{LP_1}$



4.1 Branch-and-bound method

After considering X_1, the best feasible (integer) solution found so far is:

$$x' = \begin{bmatrix} 3 \\ 3 \end{bmatrix} \text{ with } z' = 39.$$

N.B. it's a max problem

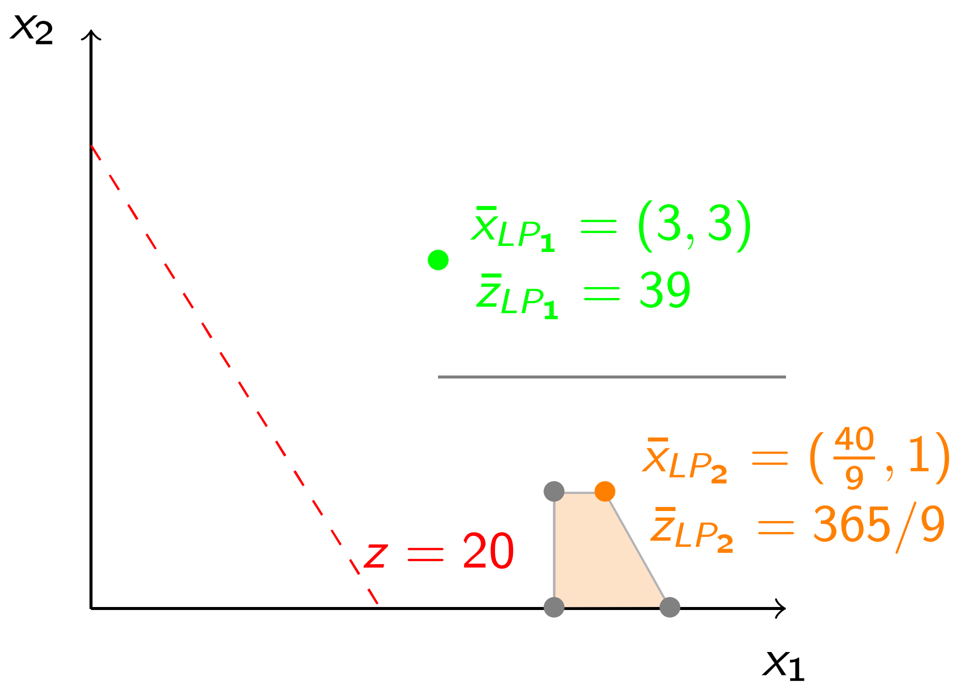
Since $z_{LP_2} = 41 > 39$ and $x_{LP_2} = \begin{bmatrix} 4 \\ 9/5 \end{bmatrix}$, X_2 may contain a better feasible solution of ILP.

$\Rightarrow$ Partition X_2 into X_3 and X_4 by imposing:

$$x_2 \leq \lfloor \bar{x}_2 \rfloor = 1 \quad \text{or} \quad x_2 \geq \lfloor \bar{x}_2 \rfloor + 1 = 2$$

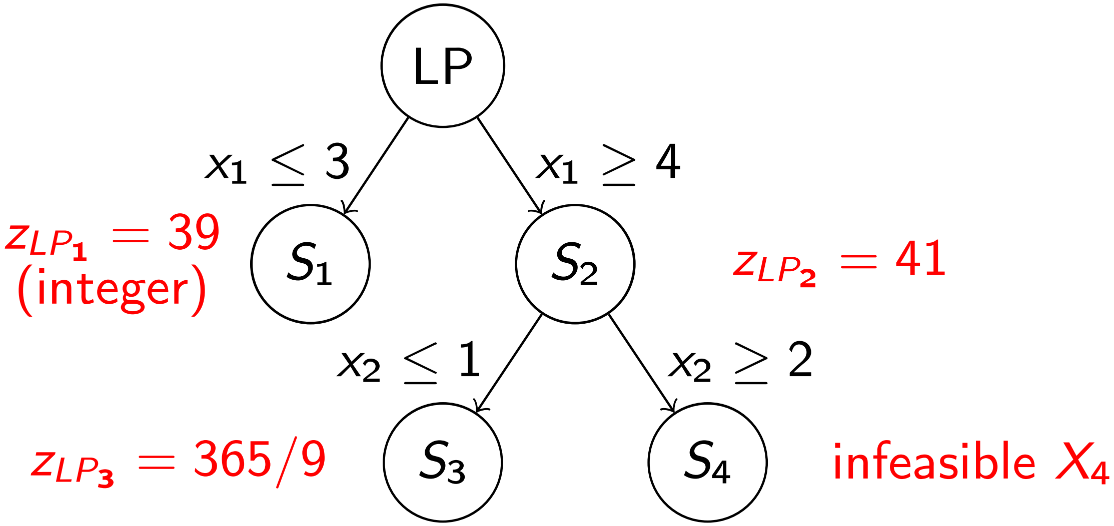
Subproblem S_3: subregion X_3

Subproblem S_4 is infeasible: subregion $X_4 = \emptyset$



4.1 Branch-and-bound method

Branching tree



Since $z_{LP_3} = 365/9 > 39$ and $\bar{x}_{LP_3} = (40/9, 1)$, X_3 may contain a better feasible solution of ILP.

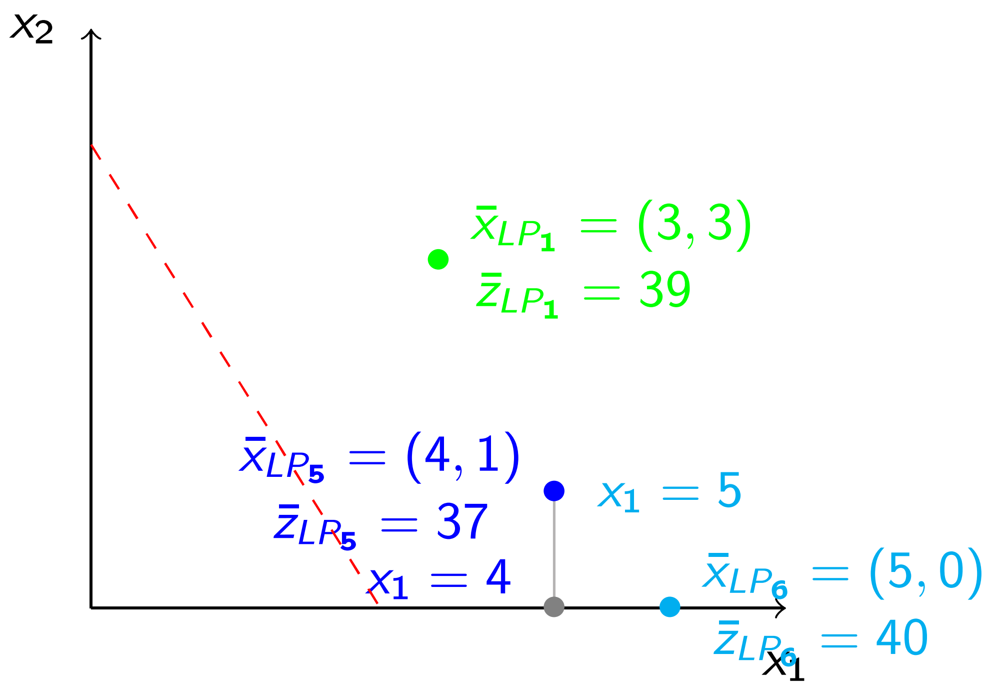
$\Rightarrow$ Partition X_3 into X_5 and X_6 by imposing:

$$x_1 \leq \lfloor x_1 \rfloor = 4 \text{ or } x_1 \geq \lfloor x_1 \rfloor + 1 = 5$$

4.1 Branch-and-bound method

Integer solution $\bar{x}_{LP_5}$ (feasible for ILP), $x' = (3,3)$, but with worse objective function value of $z' = 39$.

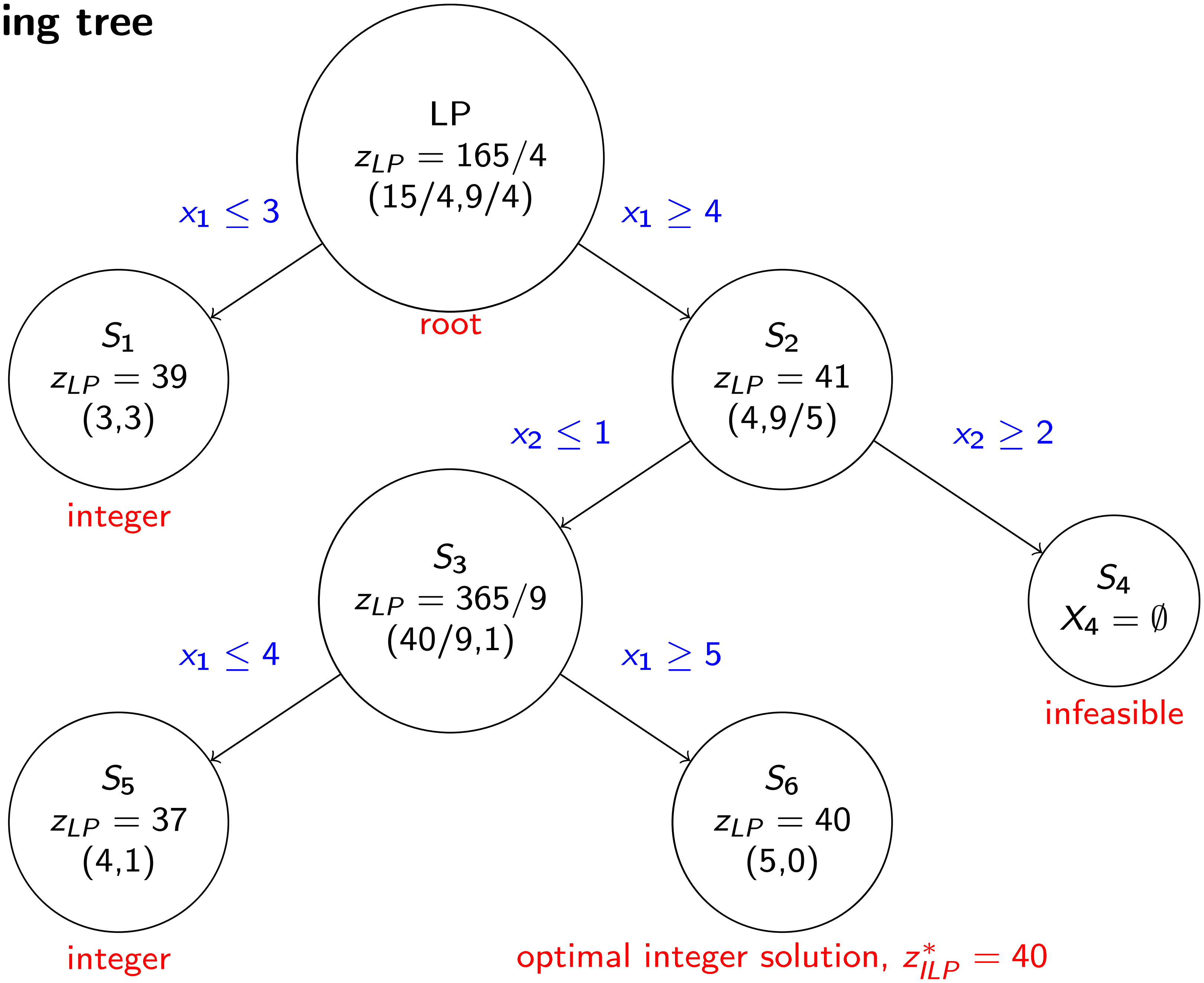
$\bar{x}_{LP_6}$ is the best integer solution found, therefore it is an optimal solution.



Branch-and-bound is an exact method (it guarantees an optimal solution).

4.1 Branch-and-bound method

Branching tree



4.1 Branch-and-bound method

The branching tree may not contain all possible nodes (2^d leaves).

A node of the tree has no child – is "fathomed" – if:

1. initial constraints + those on the arcs from the root are infeasible (e.g. S_4)
2. optimal solution of the linear relaxation is integer (e.g. S_1)
3. the value $c^T \bar{x}_{LP}$ of the optimal solution $\bar{x}_{LP}$ of the linear relaxation is worse than that of the best feasible solution of ILP found so far.

$\equiv$ Bounding criterion

Note: In case 3. the feasible subregion of the subproblem associated to that node cannot contain an integer feasible solution that is better than the best feasible solution of ILP found so far!

Bounding criterion often allows to "discard" a large number of nodes (subproblems).

4.1 Branch-and-bound method

Choice of the node (subproblem) to examine:

- Deeper nodes first (depth-first search strategy)
  Simple recursive procedure, it is easy to reoptimize but it may be costly in case of wrong choice.
- First more promising nodes – with the best linear relaxation value (best-bound first strategy)
  Typically generates a smaller number of nodes but suproblems are less constrained $\Rightarrow$ takes longer to find a first feasible solution and improve it.

Choice of the (fractional) variable for branching

- It may not be the best choice to select the variable x_h whose fractional value is closer to 0.5 (hoping to obtain two subproblems that are more stringent and balanced).
- Strong branching: Try to branch on some of candidate variables (fractional basic ones), evaluate the corresponding objective function values and actually branch on the variable that yields the best improvement in the objective function.

4.1 Branch-and-bound method

Efficient solution of the linear relaxations

- No need to solve the linear relaxations of the ILP subproblems from scratch with the two-phase Simplex algorithm.
- An optimal solution of the linear relaxation with a single additional constraint can be found via a single iteration of the Dual simplex method ($\equiv$ Simplex applied to the dual) to the optimal Branch-and-bound of the previous linear relaxation.

is another version of the simplex method

Thus, it can be done very efficiently

Remarks on Branch-and-bound method

- Branch-and-bound is also applicable to mixed ILPs: when branching just consider the fractional variables that must be integer.
- Finding a good initial feasible solution with a heuristic may improve the method's efficiency by providing a better lower bound z' on z_{ILP} (for maximization problem).

Note: Branch-and-bound can also be used as a heuristic by limiting the computing time or the number of nodes that are examined.

4.1 Branch-and-bound method

Applicability of Branch-and-bound approach
General method that can be adapted to tackle any discrete optimization problem and many nonlinear optimization problems.

e.g., scheduling, traveling salesman problem, . . .

We "just" need:

- Technique to partition a set of feasible solutions into two or more subsets of feasible solutions (branch).
- Procedure to determine a bound on the cost of any solution in such a subset of feasible solutions (bound). so far we did it by solving the linear relaxation.

4.1.2 Branch-and-bound for a combinatorial optimization problem

Example: Scheduling problem on a single machine (NP-hard)
Given n jobs with deadlines, and a single machine with processing time for each job, determine a sequence that minimizes the total delay.

jobs	processing time (h)	deadline (completed by hour)
1	6	8
2	4	4
3	5	12
4	8	16

The sequence 1–2–3–4, has a total delay $= 0 + 6 + 3 + 7 = 16$ hours.

4.1 Branch-and-bound method

Define $x_{ij} = \begin{cases} 1 \text{ if job } i \text{ is processed in } j\text{-th position} \\ 0 \text{ otherwise} \end{cases}$

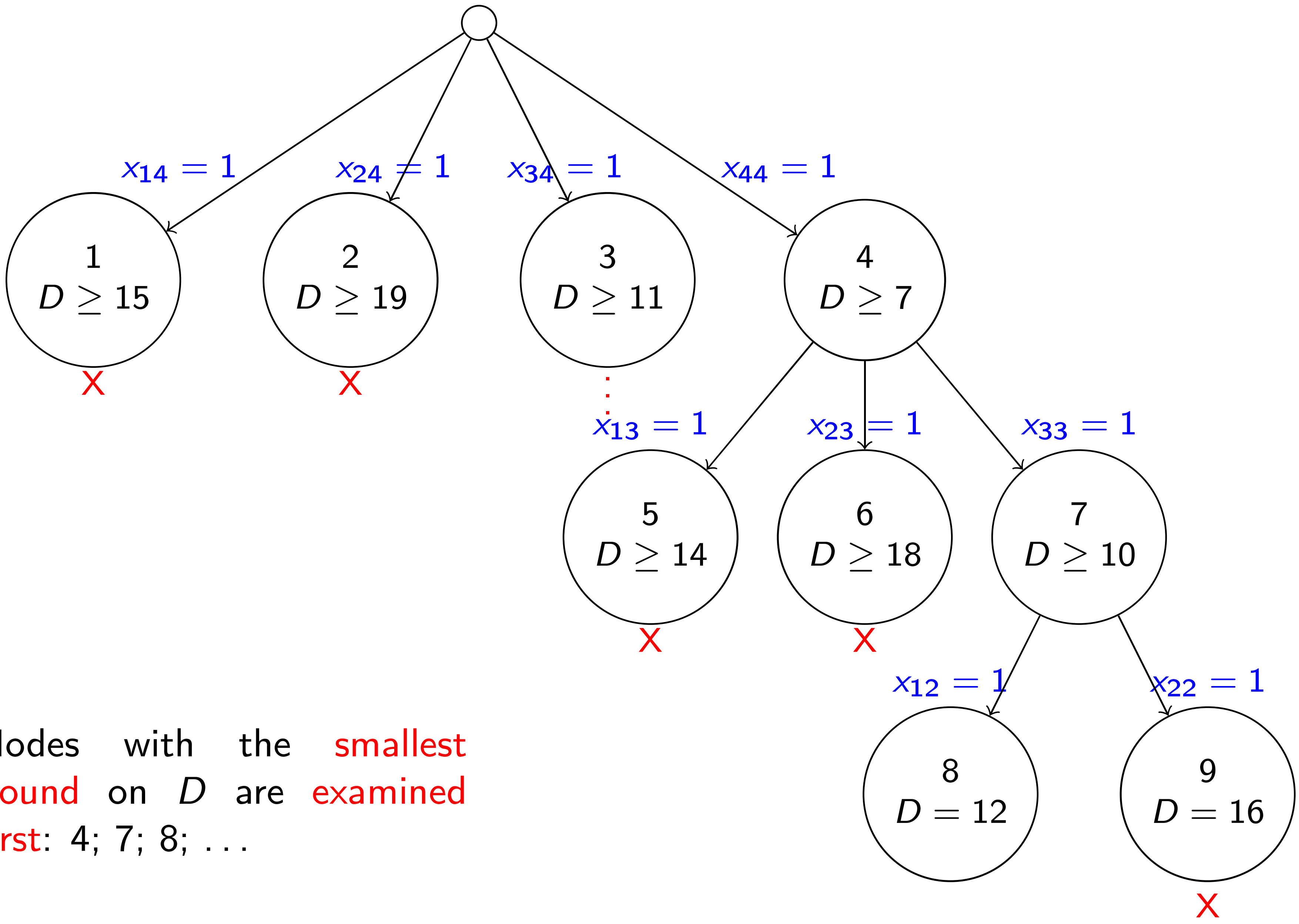
Idea of the method:
Partition the set of all feasible solutions based on the last processed job.

In this case, either $x_{14} = 1$ or $x_{24} = 1$ or $x_{34} = 1$ or $x_{44} = 1$.

Let D be the total delay.

- Processing all the jobs requires $6 + 4 + 5 + 8 = 23$ hours.
- If $x_{44} = 1$, job 4 is completed at the end of hour 23, therefore with a delay of $23 - 16 = 7$.
- Thus, this is a lower bound for the delay of any solution with job 4 the last job to be processed ($D \geq 7$).

4.1 Branch-and-bound method



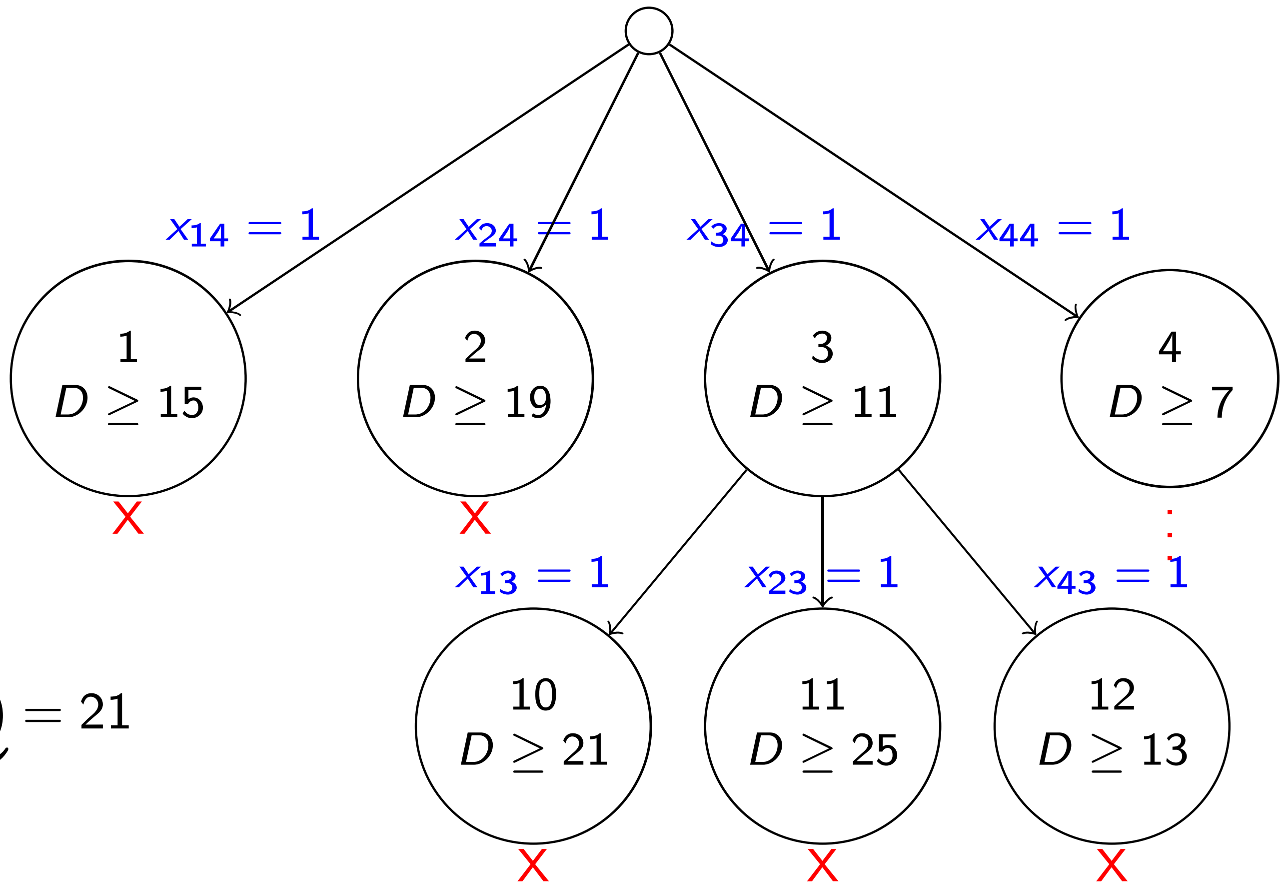
Nodes with the smallest bound on D are examined first: 4; 7; 8; . . .

4.1 Branch-and-bound method

- For node 7:
  - Job 4 is the last one with a delay of 7 hours
  - Job 3 is the next to last one with a delay of $6 + 4 + 5 - 12 = 15 - 12 = 3$ hours $\Rightarrow D \geq 7 + 3 = 10$
- Node 8: Sequence 2–1–3–4 is a feasible solution with delay 12

Note: Nodes 1, 2, 5 and 6 can be "pruned"!

4.1 Branch-and-bound method



$$11 + \underbrace{(6 + 4 + 8 - 8)}_{\text{job 1 next to last}} = 21$$

$$11 + \underbrace{(6 + 4 + 8 - 16)}_{\text{job 4 next to last}} = 13$$

$\Rightarrow$ Optimal sequence: 2–1–3–4 with $D = 12$.

4.1 Branch-and-bound method

Apply a branch-and-bound method to solve the knapsack problem:

$$\begin{array}{ll} \max & 8x_1 + 5x_2 + 5x_3 + 3x_4 + x_5 \\ \text{s. t.} & 4x_1 + 3x_2 + 4x_3 + 3x_4 + 2x_5 \leq 12 \\ & x_1, x_2, x_3, x_4, x_5 \in \{0, 1\} \end{array}$$

To solve the linear programming relaxation of the problem:

$$\begin{array}{ll} \max & 8x_1 + 5x_2 + 5x_3 + 3x_4 + x_5 \\ \text{s. t.} & 4x_1 + 3x_2 + 4x_3 + 3x_4 + 2x_5 \leq 12 \\ & x_1, x_2, x_3, x_4, x_5 \in [0, 1] \end{array}$$

8/4 >= 5/3 >= ...

we use the following greedy algorithm.

1. Sort the variables by nonincreasing c_i/a_i. (In this case, they are already ordered appropriately.)
2. Following that order, assign the largest possible value to the variable under consideration $x_{i'}$ as long as $\sum_{i<i'} c_i \leq B$:
   $x_{i'} = 1$ if $\sum_{i \leq i'} c_i \leq B$; $x_{i'} = \frac{B - \sum_{i<i'} c_i}{c_{i'}}$ if $\sum_{i<i'} c_i + c_{i'} > B$; $x_i = 0$, $i > i'$.

4.1 Branch-and-bound method

The optimal solution for the linear relaxation of the problem is $x^*_{LR} = (1, 1, 1, \frac{1}{3}, 0)$, $z_{UB} = 19$. x_4 is fractional, so we branch on x_4, imposing $x_4 = 1$ or $x_4 = 0$.

P1. Imposing $x_4 = 1$, the optimal solution of the linear relaxation is $x^*_{LR} = (1, 1, \frac{1}{2}, 1, 0)$, $z_{UB} = \lfloor \frac{37}{2} \rfloor = 18$ (because the coefficient are integers).

P2. Imposing $x_4 = 0$, the optimal solution of the linear relaxation is $x^*_{LR} = (1, 1, 1, 0, \frac{1}{2})$, $z_{UB} = \lfloor \frac{37}{2} \rfloor = 18$ (because the coefficient are integers).

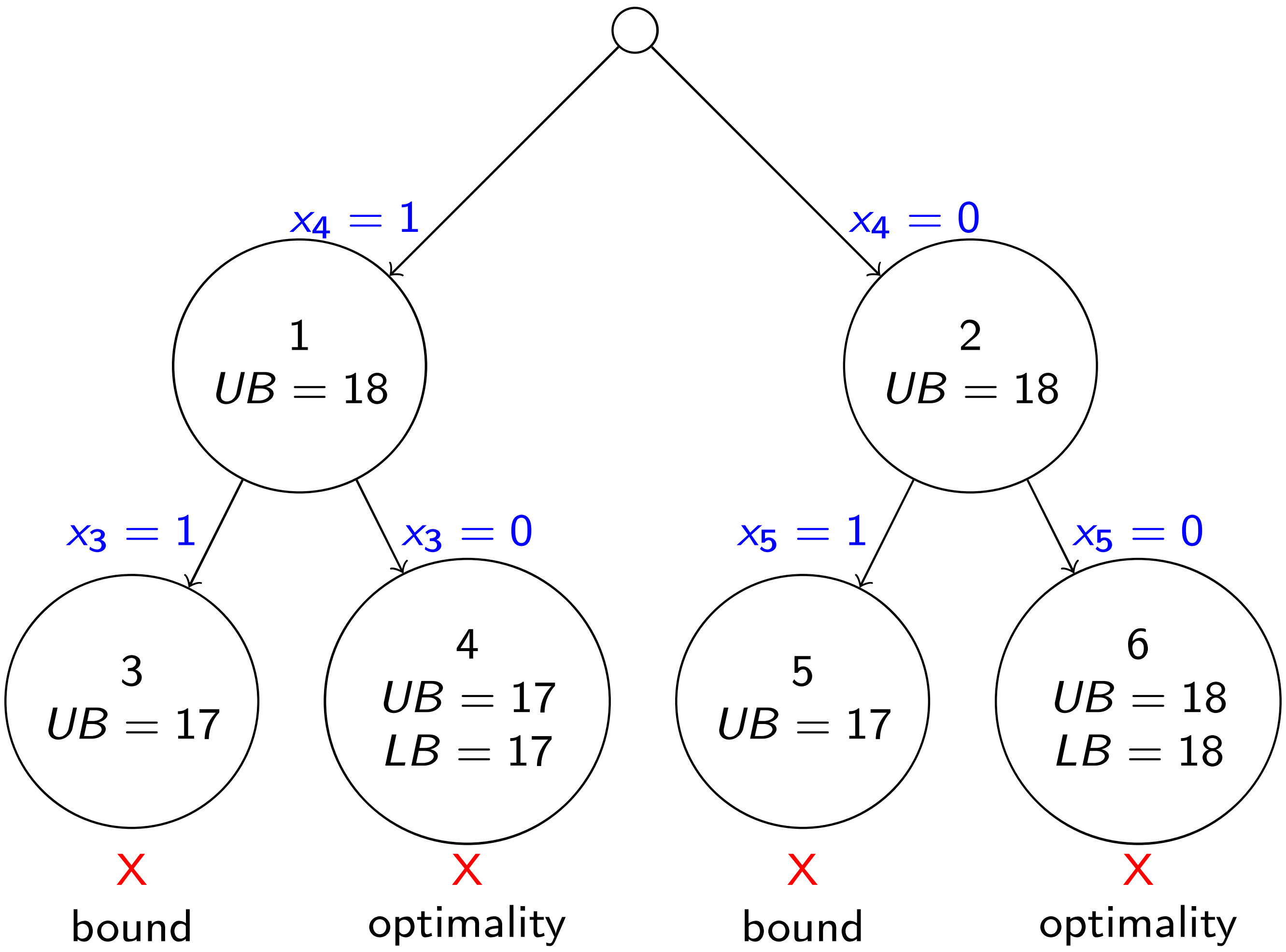
P3. We branch P1 on x_3, imposing $x_4 = x_3 = 1$. Then $x^*_{LR} = (1, \frac{1}{3}, 1, 1, 0)$, $z_{UB} = \lfloor \frac{53}{3} \rfloor = 17$ (because the coefficient are integers).

P4. We branch P1 on x_3, imposing $x_4 = 1$, $x_3 = 0$. Then $x^*_{LR} = (1, 1, 0, 1, 1)$, $z_{UB} = z_{LB} = \lfloor \frac{53}{3} \rfloor = 17$ because this solution is feasible. This node is pruned by optimality whereas node P3 is pruned by bound.

P5. We branch P2 on x_5, imposing $x_4 = 1$, $x_5 = 0$. Then $x^*_{LR} = (1, 1, \frac{3}{4}, 0, 1)$, $z_{UB} = \lfloor \frac{71}{4} \rfloor = 17$ (because the coefficient are integers). This node is pruned by bound.

P6. We branch P2 on x_5, imposing $x_4 = x_5 = 0$. Then $x^*_{LR} = (1, 1, 1, 0, 0)$, $z_{UB} = z_{LB} = 18$ because this solution is feasible. This node is pruned by optimality.

4.1 Branch-and-bound method



Then $x^*_{LR} = (1, 1, 1, 0, 0)$ is an optimal solution, $z^* = 18$.