BRANCH-AND-BOUND: BASIC IDEA

Integer Linear Program: Example

$$
\begin{array}{rlcl}
\max & -x_1 - 2x_2 - 0.5x_3 - 0.2x_4 - x_5 + 0.6x_6 & & \\
\text{s.t.} & x_1 + 2x_2 & \geq & 1 \\
 & x_1 + x_2 + 3x_6 & \geq & 1 \\
 & x_1 + x_2 + x_6 & \geq & 1 \\
 & x_3 - 3x_4 & \geq & 1 \\
 & x_3 - 2x_4 - 5x_5 & \geq & 1 \\
 & x_4 + 3x_5 - 4x_6 & \geq & 1 \\
 & x_2 + x_5 + x_6 & \geq & 1 \\
0 \leq & x_1, x_2, \cdots, x_6 & \leq & 10 \\
 & x_1, \ldots, x_6 & \in & \mathbb{Z}
\end{array}
$$

Step #1: Solve the LP relaxation

Solving the LP relaxation.

```
x1.val = 0.3333333333333333
x2.val = 0.6666666666666667
x3.val = 2.666666666666667
x4.val = 0
x5.val = 0.3333333333333333
x6.val = 0

Optimal Value (LP relaxation): -3.333333
```

Step #2: Choose a branch variable

```
x1.val = 0.3333333333333333
x2.val = 0.6666666666666667
x3.val = 2.6666666666666667
x4.val = 0
x5.val = 0.3333333333333333
x6.val = 0

Optimal Value (LP relaxation): -3.333333
```

Choose a variable that should be integer

Step #3: Branching Constraints

$$
\begin{array}{lll}
\max & -x_1 - 2x_2 - 0.5x_3 - 0.2x_4 - x_5 + 0.6x_6 & \\
\text{s.t.} & x_1 + 2x_2 & \geq 1 \\
& x_1 + x_2 + 3x_6 & \geq 1 \\
& x_1 + x_2 + x_6 & \geq 1 \\
& x_3 - 3x_4 & \geq 1 \\
& x_3 - 2x_4 - 5x_5 & \geq 1 \\
& x_4 + 3x_5 - 4x_6 & \geq 1 \\
& x_2 + x_5 + x_6 & \geq 1 \\
0 \leq & x_1, x_2, \cdots, x_6 & \leq 10
\end{array}
$$

Original Problem

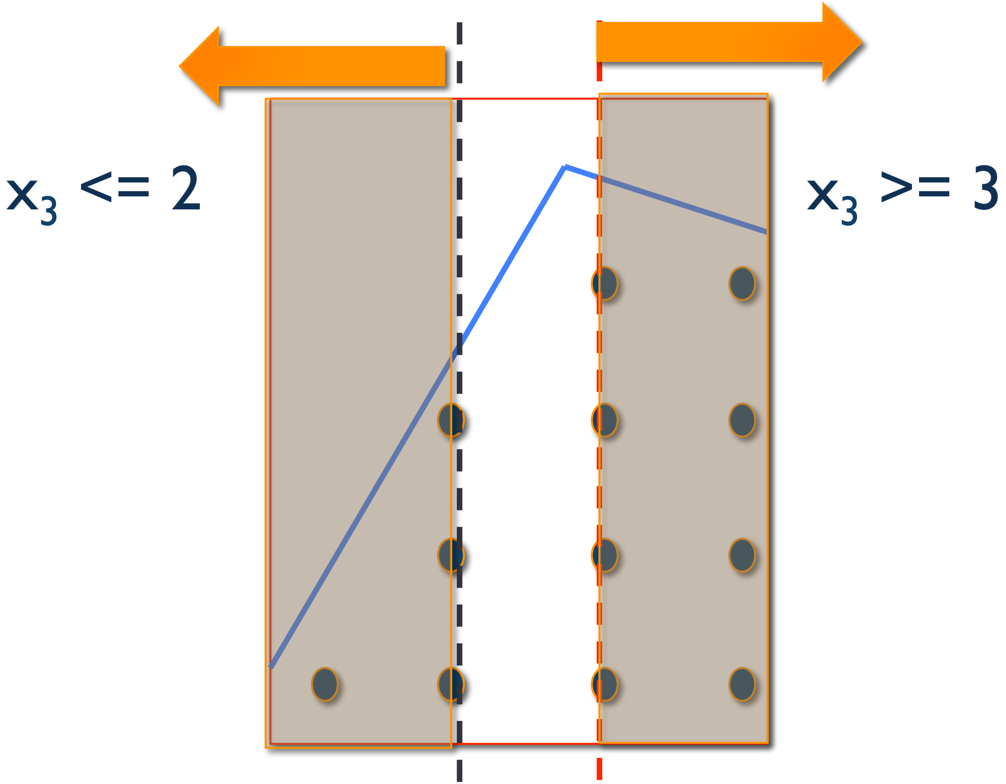
b1

b2

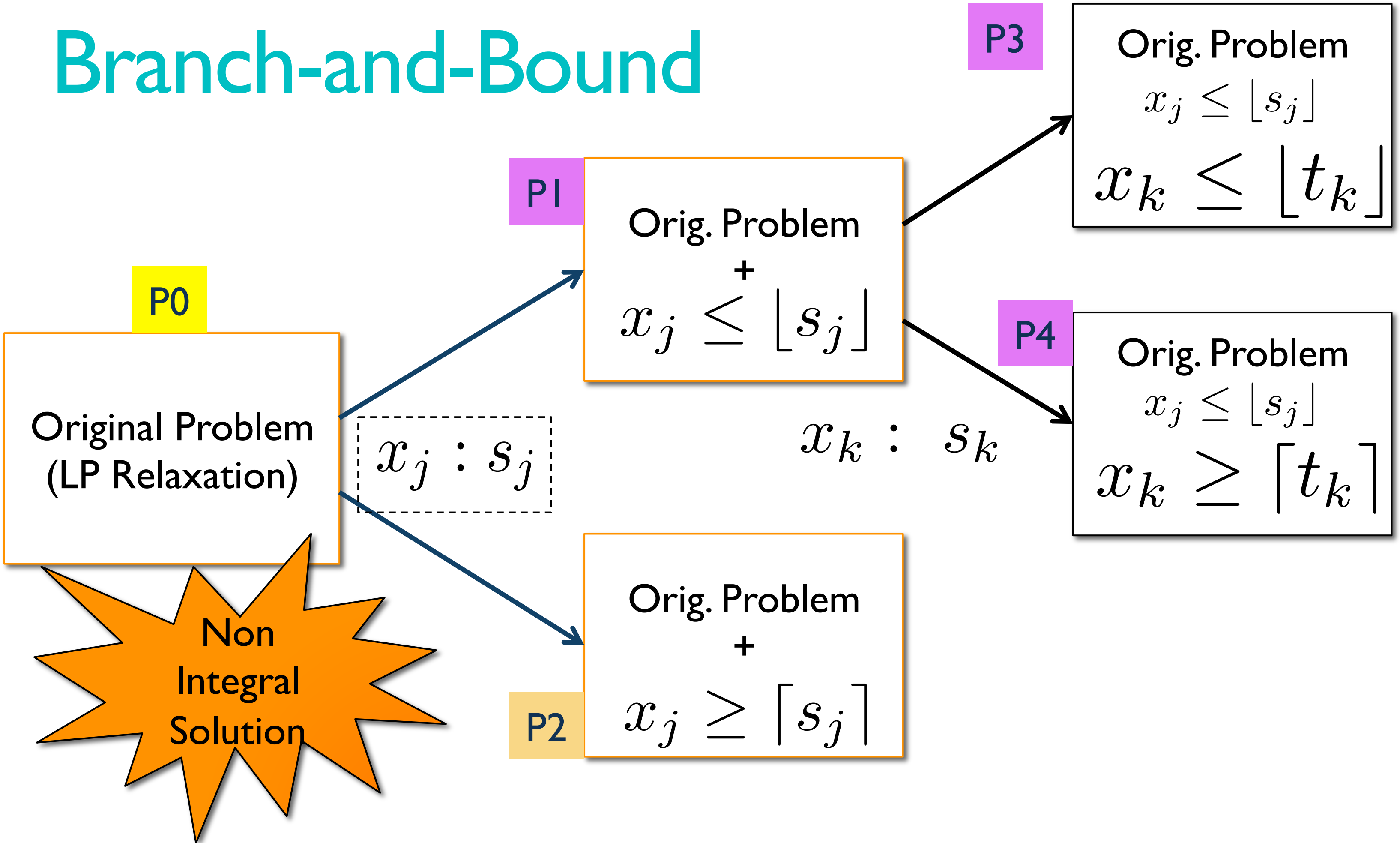
Original Problem Constr.
+
(x_3 <= 2)

Original Problem Constr.
+
(x_3 >= 3)

Visualization

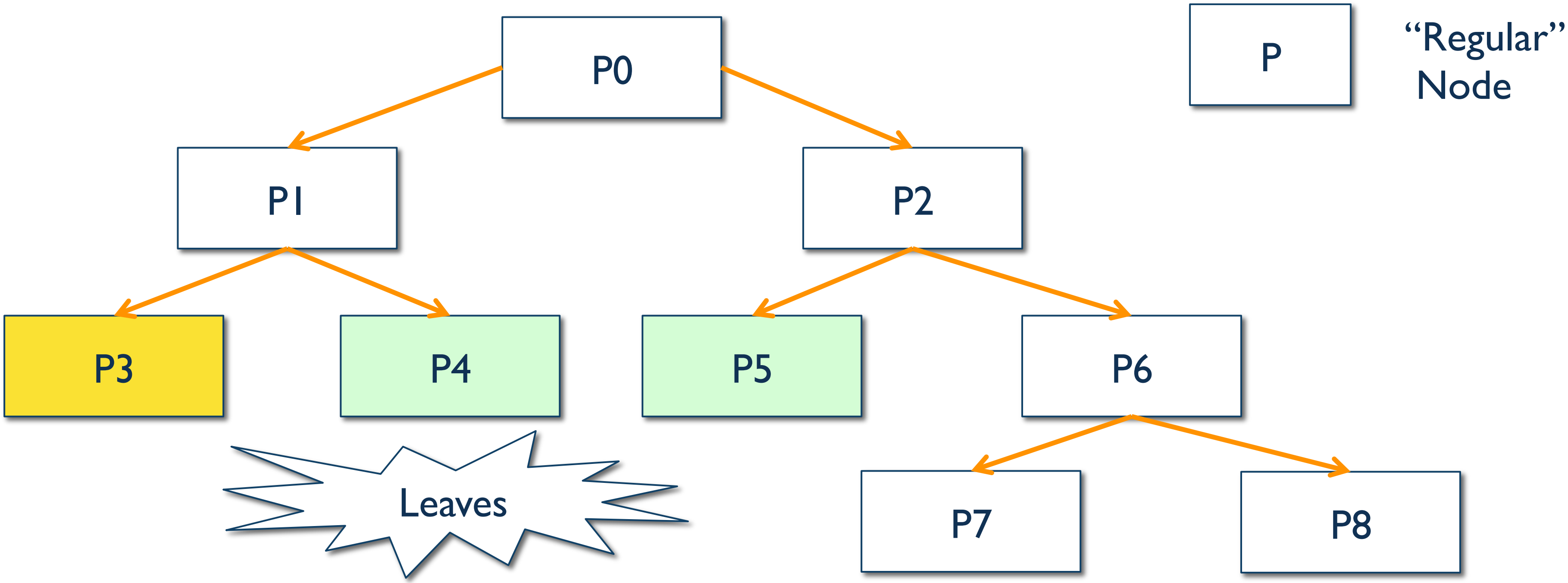


Branch-and-Bound



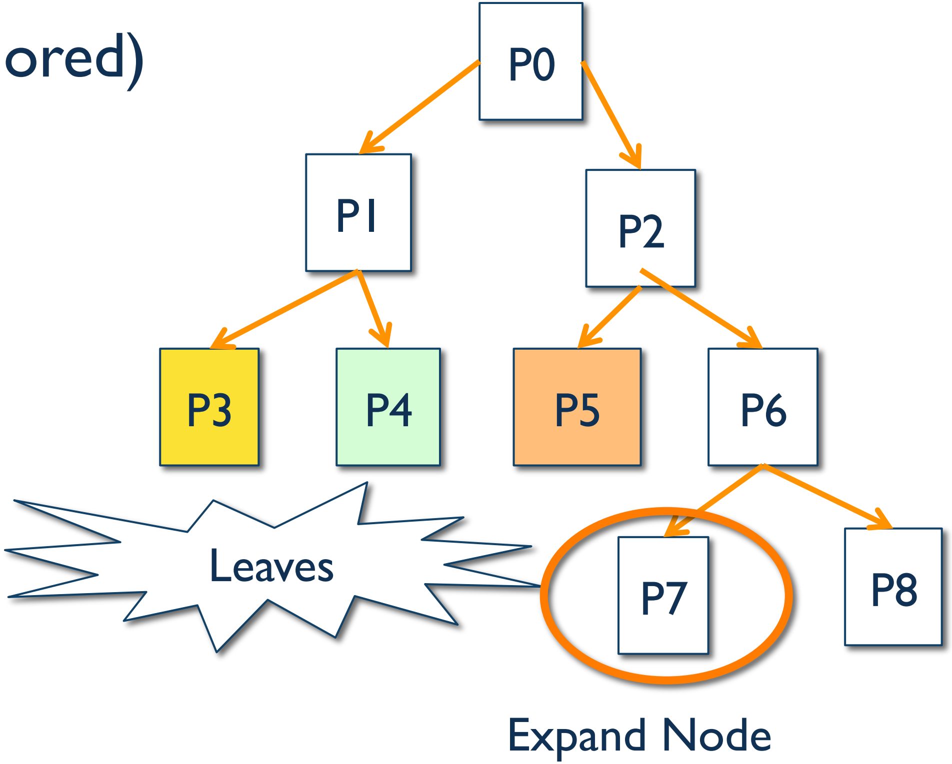
Branch-And-Bound: Operations

Branch-And-Bound Tree: Nodes are ILP instances.



Branch-And-Bound Tree

P0	Regular Node (to be explored)
P3	Leaf node
P4	Leaf node
P5	Leaf node



Expanding a Node

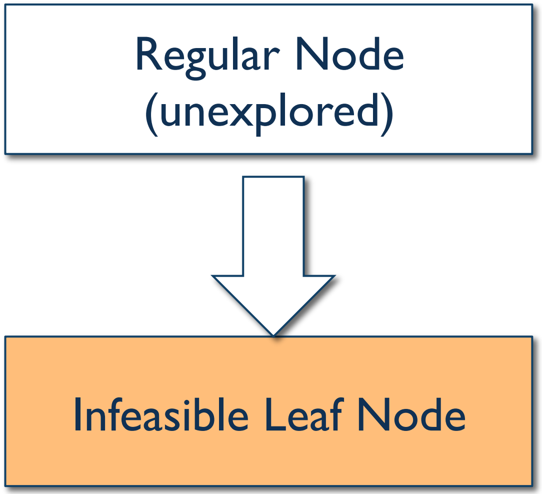
1. Solve the LP relaxation for the node ILP.

Regular Node (unexplored)	LP relaxation solution

2. Three cases:
   1. Infeasible.
   2. Integral Solution.
   3. Fractional Solution.

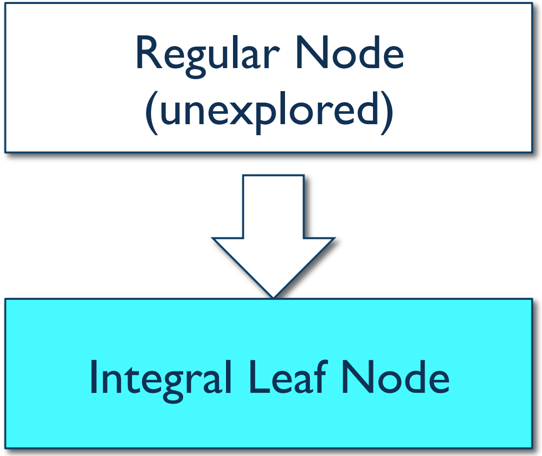
Case-1: Infeasible LP relaxation.

- LP relaxation is infeasible.



Case-2: LP relaxation yields integral solution

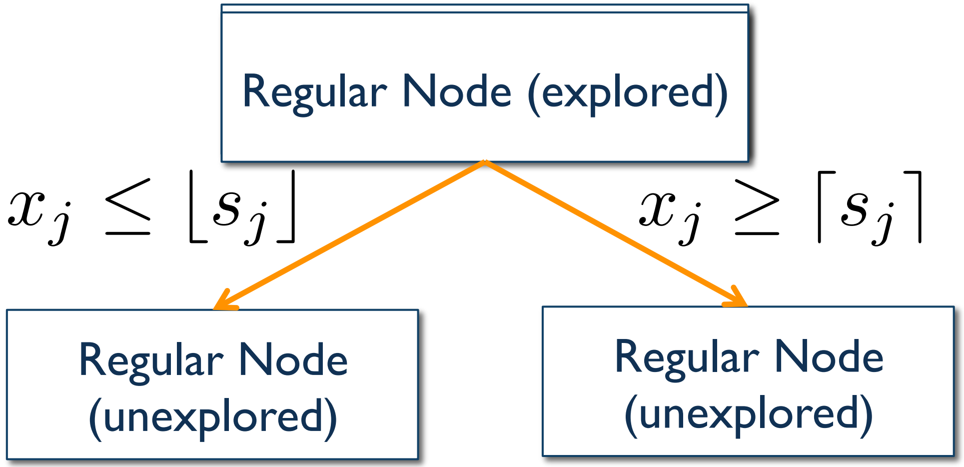
- LP relaxation solution is integral: ILP solution = LP solution.



bestObjective := max (lpOptimum, bestObjective)

Case-3: LP relaxation yields a fractional solution

- LP relaxation yields a fractional solution.



LP relaxation solution:

x1: ...

x2: ...

x3: ...

...

Opt. Solution: optSolution

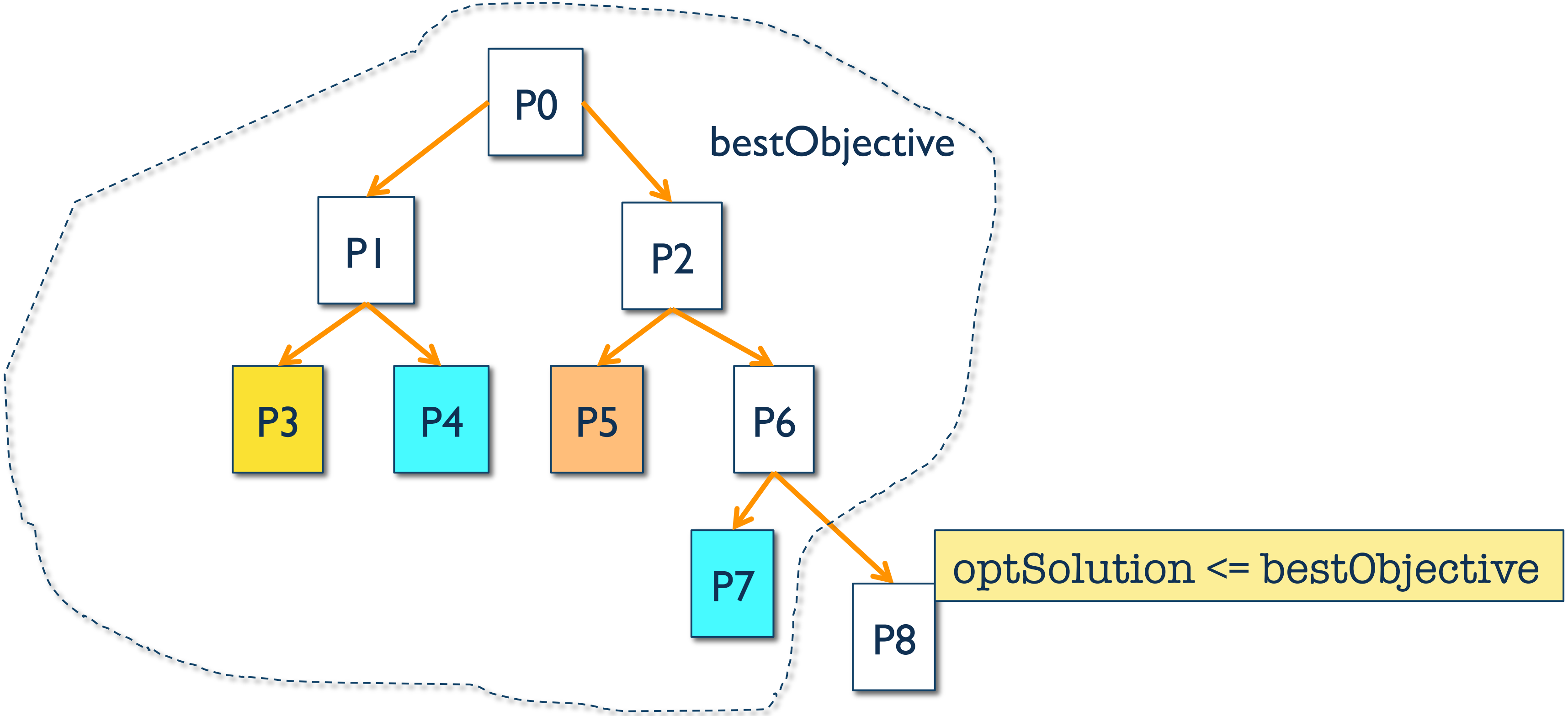
Optimal Pruning

Regular Node
(explored)

optSolution <= bestObjective?

LP relaxation solution:
x1: ...
x2: ...
x3: ...
...
Opt. Solution: optSolution

Optimal Pruning



Branch-And-Bound Initial Node

Original ILP
(unexplored
node)

`bestObjective := -Infinity`