

# Solution Methods for Mixed-Integer Optimization Problems :

## The Branch-and-Bound Algorithm

# Example

$$\begin{array}{ll}\max & 2x_1 + 5x_2 \\ \text{s.t.} & 5x_1 + 2x_2 \leq 81 \\ & 2x_1 + 8x_2 \leq 80 \\ & x_1, x_2 \geq 0, \text{ integer}\end{array}$$

Integer program

$$\begin{array}{ll}\max & 2x_1 + 5x_2 \\ \text{s.t.} & 5x_1 + 2x_2 \leq 81 \\ & 2x_1 + 8x_2 \leq 80 \\ & x_1, x_2 \geq 0, \text{ integer}\end{array}$$

Linear Programming (LP) relaxation

# Branch-and-Bound: bounding

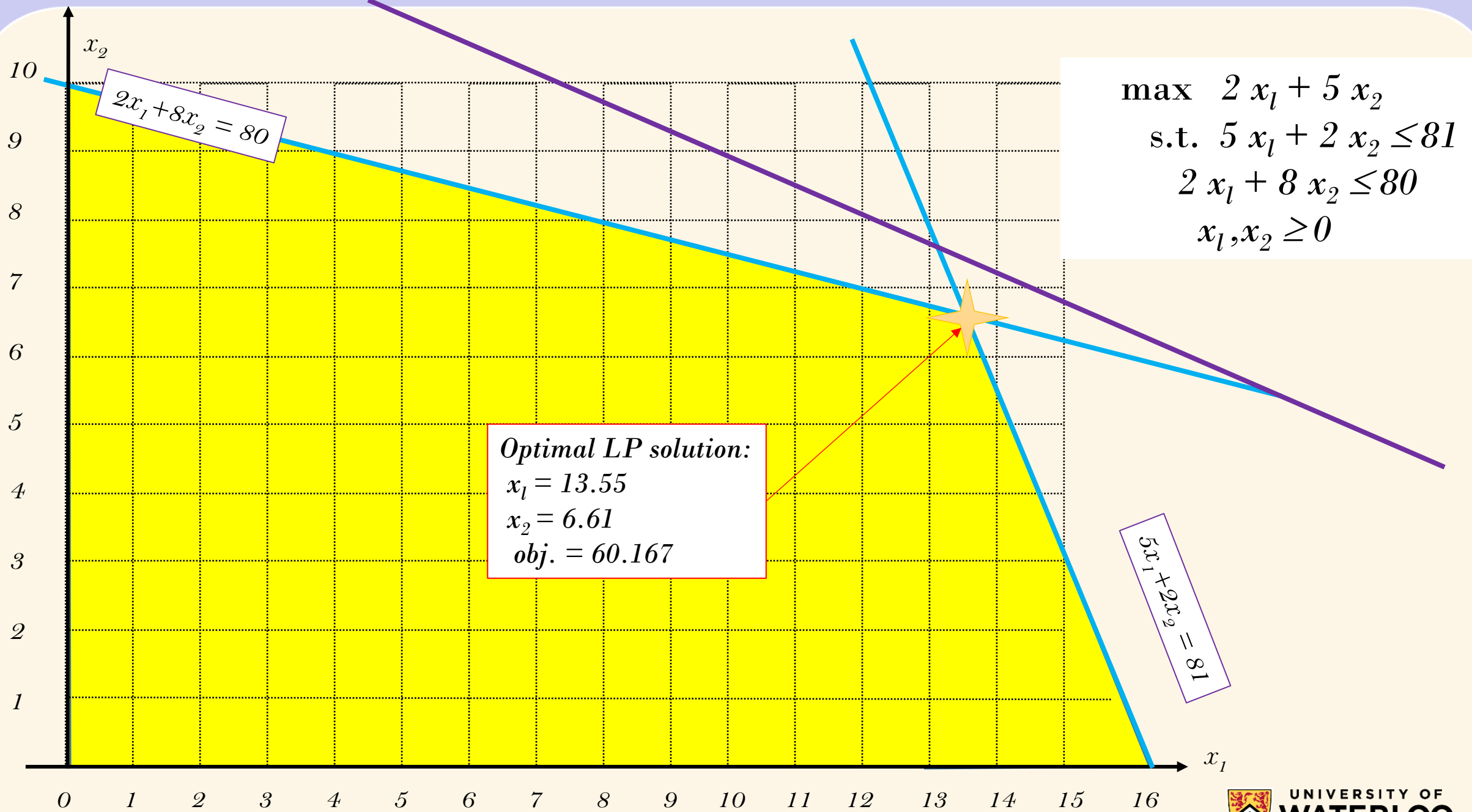
For a minimization (maximization) problem, lower (upper) bounding can be achieved through Relaxation (ignoring some of the constraints/requirements)

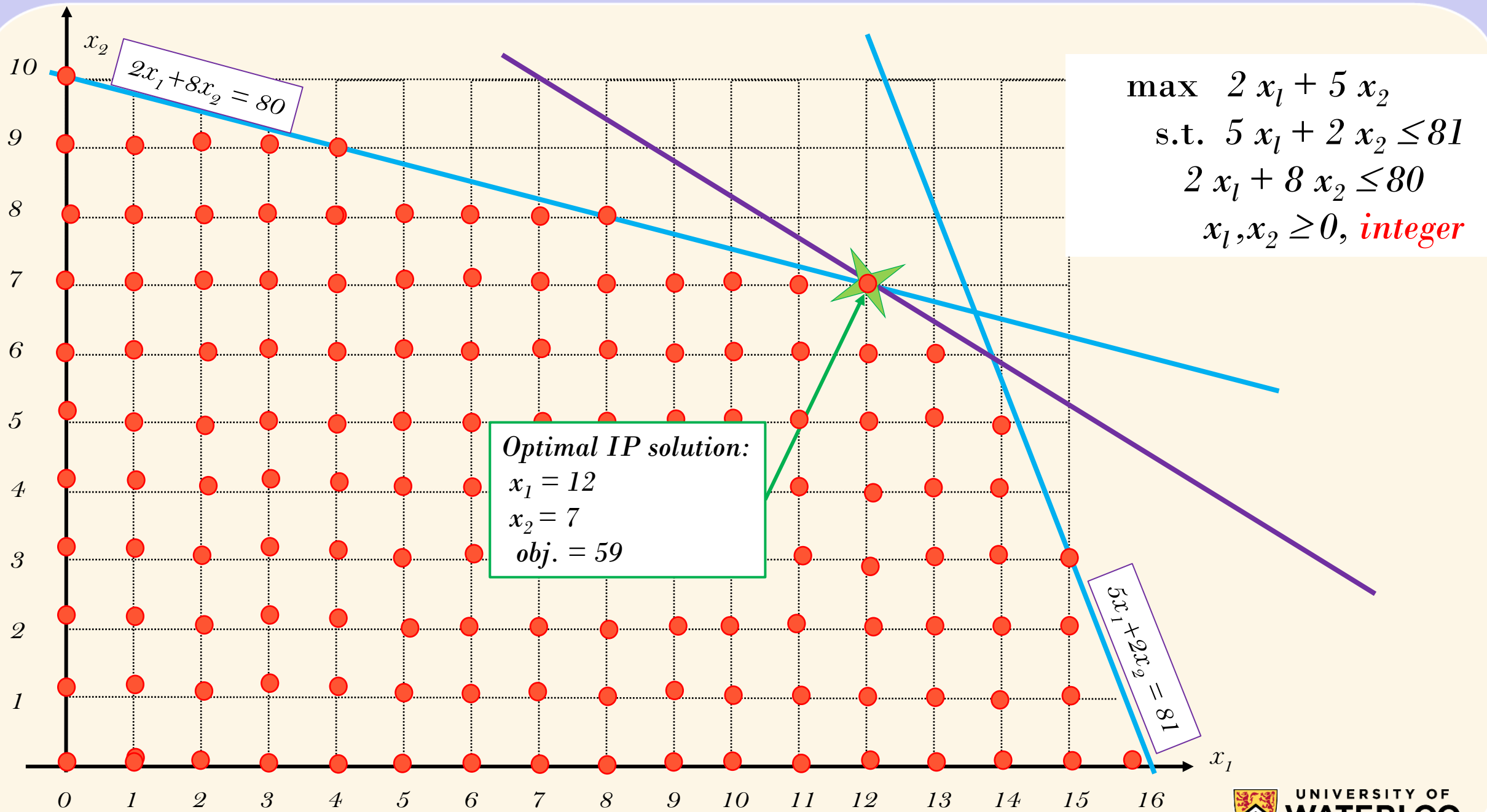
- LP relaxation: ignores integer requirements
- Constraint (Combinatorial) relaxation: ignores some constraints
- Lagrangean relaxation

Upper (lower) bounding is achieved through feasible solutions

- Using a heuristic
- is not needed for Branch-and-bound, but if used will enhance the performance

$$\begin{array}{ccccc} \text{Lower} & & \text{Optimal objective of a} & & \text{Upper} \\ \text{bound} & \leq_{\text{Relaxation}} & \text{problem to be minimized} & \leq_{\text{Heuristics}} & \text{bound} \end{array}$$





# The Branch-and-Bound algorithm

Consider

$$z = \left\{ \max \sum_{j=1}^m c_j x_j = c^T x : x \in S \right\}$$

**Idea:** If a problem is hard, partition its feasible region  $S$  into subsets  $S_k$ , and solve each of the smaller subproblems:

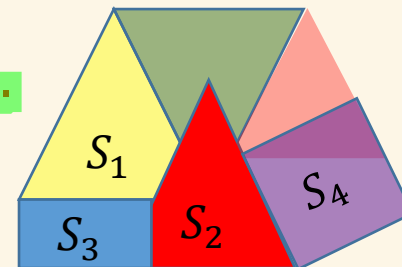
$$z^k = \{ \max c^T x : x \in S_k \}$$

Let  $S = S_1 \cup S_2 \cup \dots \cup S_k$  be a partition of the set  $S$ , and let

$$z^k = \{ \max c^T x : x \in S_k \} \text{ for } k = 1, \dots, K.$$

then

$$z = \max_k z^k.$$



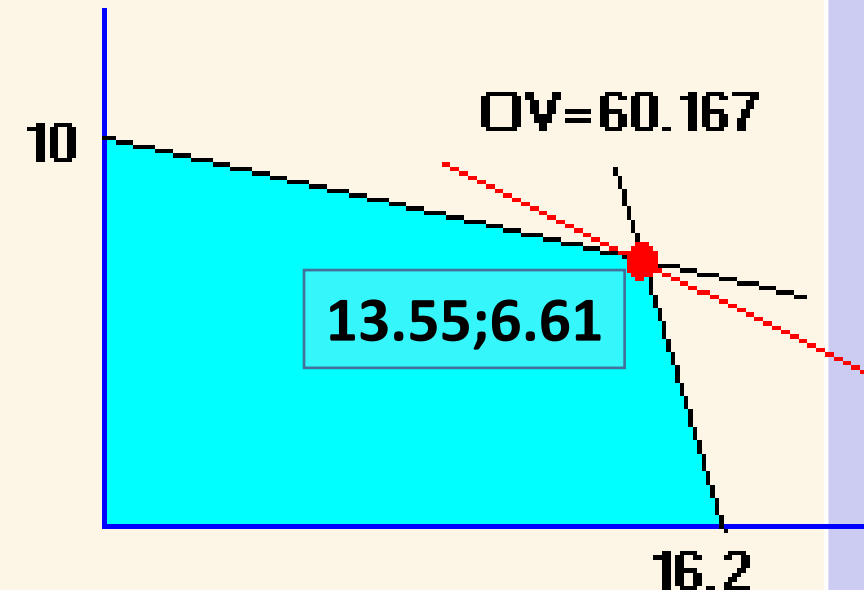
## The Branch-and-Bound algorithm in action

The Branch-and-bound algorithm relies on the strategy of separating (branching) the problem into sub-problems, each having a smaller feasible region, and calculating bounds on the best solution that can be obtained from these smaller problems.

### 1<sup>st</sup> step: Solve the corresponding Linear Program

$$\begin{array}{ll}\text{LP:} & \max \quad 2x_1 + 5x_2 \\ & \text{s.t.} \quad 5x_1 + 2x_2 \leq 81 \\ & \quad \quad 2x_1 + 8x_2 \leq 80 \\ & \quad \quad x_1, x_2 \geq 0, \text{ integer}\end{array}$$

The solution is  $x_1 = 13.55$ ,  $x_2 = 6.61$  with an objective value (OV) of 60.167.



## Rounding:

rounding the solution to the nearest integer produces the **infeasible** point  $x_1=14, x_2=7$ .

If, however, the solution is rounded to  $x_1=13, x_2=6$ , this produces a feasible integer point. The Objective value evaluated at this point gives the first value of the **incumbent, the best known feasible solution so far**, which is 56.

$$56 \leq \text{The optimal objective} \leq 60.167$$

## 2<sup>nd</sup> step: Branching

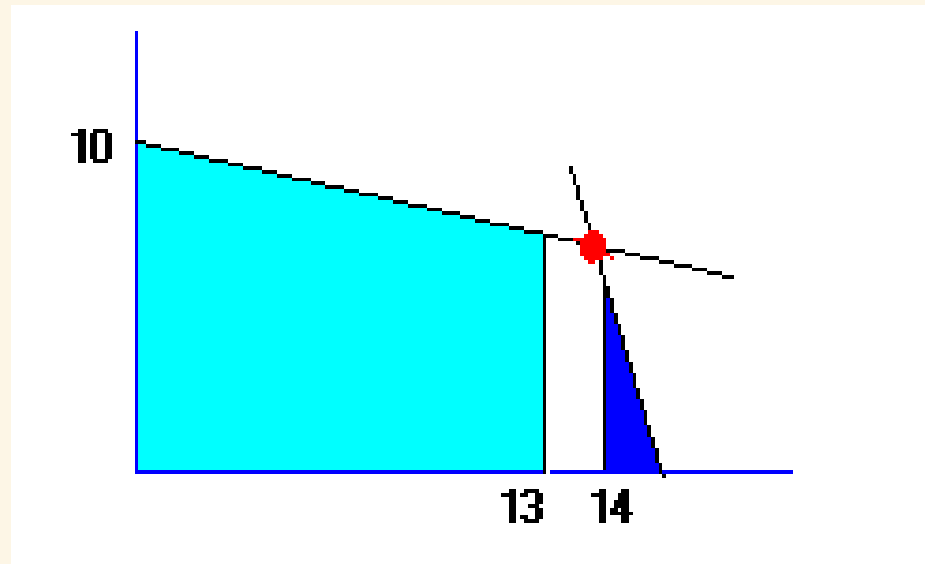
To continue the search for the optimal solution we separate the problem (**branch**) by noting that the optimal IP solution will satisfy either

$$x_1 \leq 13 \text{ or } x_1 \geq 14$$

This is done to eliminate the LP solution where  $x_1 = 13.55$   
Leads to two problems

LP 2:

$$\begin{aligned} \max \quad & 2x_1 + 5x_2 \\ \text{s.t.} \quad & 5x_1 + 2x_2 \leq 81 \\ & 2x_1 + 8x_2 \leq 80 \\ & x_1 \leq 13. \\ & x_1, x_2 \geq 0 \end{aligned}$$



LP 3:

$$\begin{aligned} \max \quad & 2x_1 + 5x_2 \\ \text{s.t.} \quad & 5x_1 + 2x_2 \leq 81 \\ & 2x_1 + 8x_2 \leq 80 \\ & x_1 \geq 14. \\ & x_1, x_2 \geq 0 \end{aligned}$$

Solve one of them

LP 2:

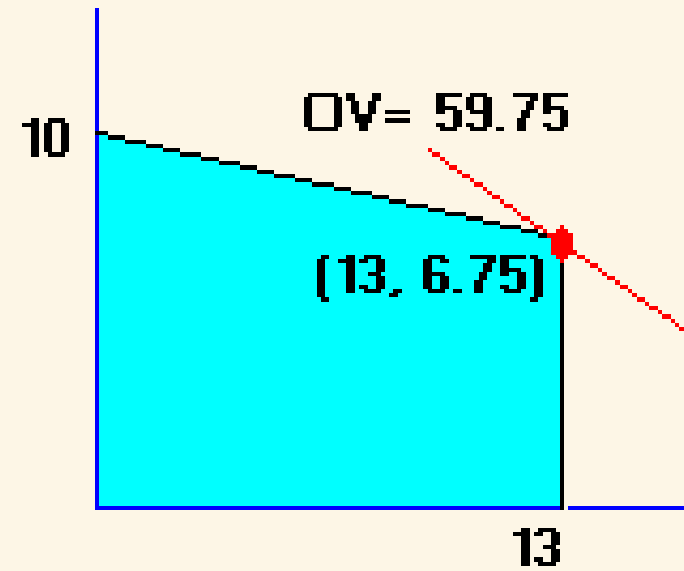
$$\max \quad 2x_1 + 5x_2$$

$$\text{s.t.} \quad 5x_1 + 2x_2 \leq 81$$

$$2x_1 + 8x_2 \leq 80$$

$$x_1 \leq 13.$$

$$x_1, x_2 \geq 0$$

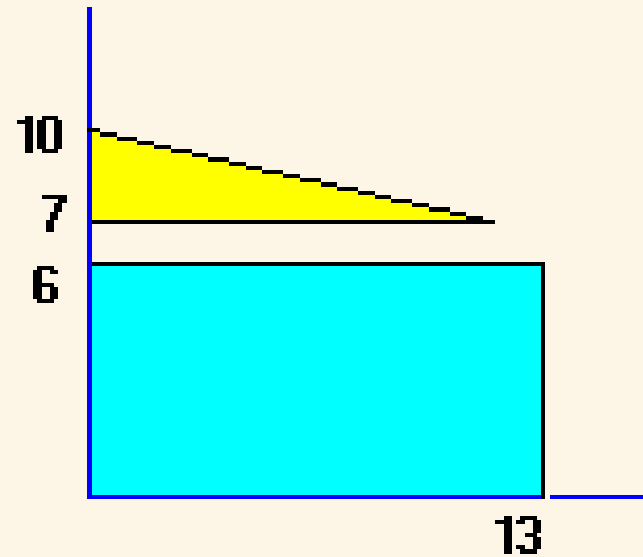


The solution is  $x_1 = 13$ ,  $x_2 = 6.75$  with an objective value (OV) of 59.75 .

Continue to branch →  
two new problems

LP4:

$$\begin{array}{ll}\max & 2x_1 + 5x_2 \\ \text{s.t.} & 5x_1 + 2x_2 \leq 81 \\ & 2x_1 + 8x_2 \leq 80 \\ & x_1 \leq 13. \\ & x_2 \leq 6 \\ & x_1, x_2 \geq 0\end{array}$$

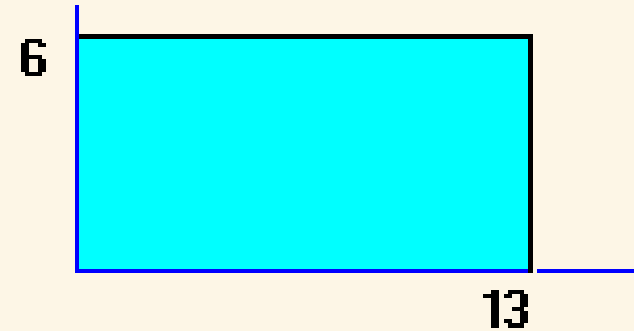


LP5:

$$\begin{array}{ll}\max & 2x_1 + 5x_2 \\ \text{s.t.} & 5x_1 + 2x_2 \leq 81 \\ & 2x_1 + 8x_2 \leq 80 \\ & x_1 \leq 13 \\ & x_2 \geq 7 \\ & x_1, x_2 \geq 0\end{array}$$

Solve LP4:

$$\begin{aligned} \text{LP4: } \max \quad & 2x_1 + 5x_2 \\ \text{s.t. } \quad & 5x_1 + 2x_2 \leq 81 \\ & 2x_1 + 8x_2 \leq 80 \\ & x_1 \leq 13 \\ & x_2 \leq 6 \\ & x_1, x_2 \geq 0 \end{aligned}$$



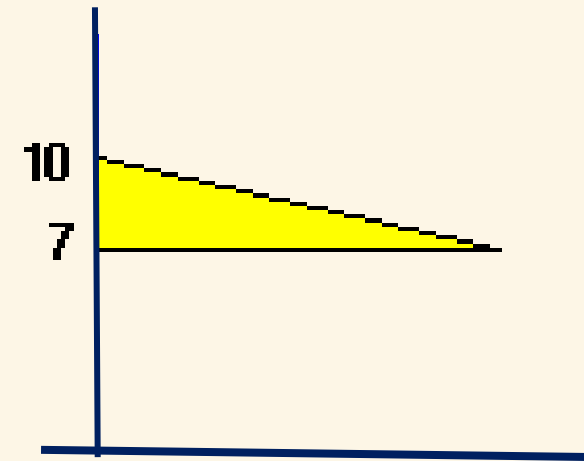
The solution is  $x_1 = 13$ ,  $x_2 = 6$  with an objective value (OV) of 56

Reached an all-integer solution

- Update incumbent if possible
- Stop exploring that node (no further branching)

Solve LP5:

$$\begin{array}{ll}\text{LP5: max} & 2x_1 + 5x_2 \\ \text{s.t.} & 5x_1 + 2x_2 \leq 81 \\ & 2x_1 + 8x_2 \leq 80 \\ & x_1 \leq 13 \\ & x_2 \geq 7 \\ & x_1, x_2 \geq 0\end{array}$$



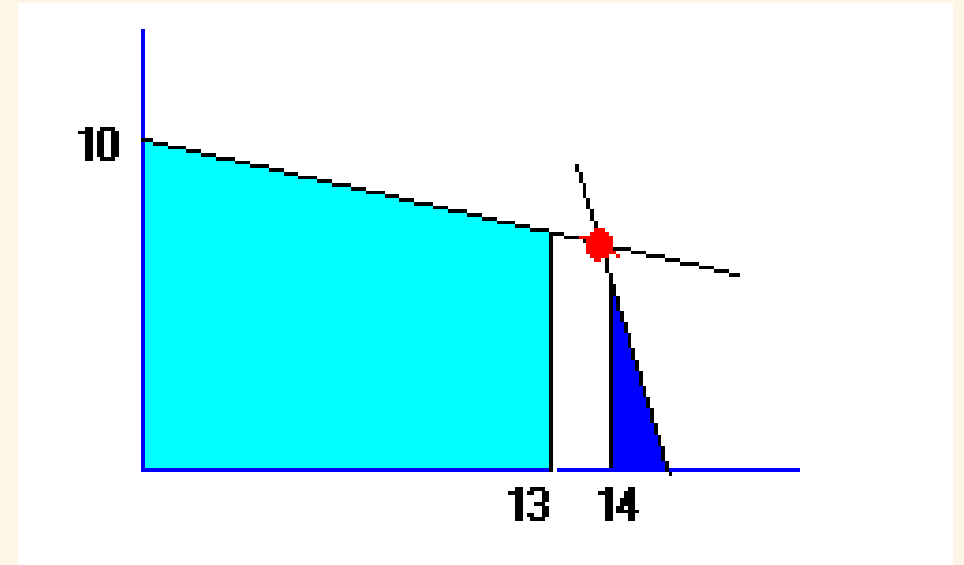
The solution is  $x_1 = 12$ ,  $x_2 = 7$  with an objective value (OV) of 59

Reached an all-integer solution

- Update incumbent:  $59 > 56 \rightarrow \text{incumbent} = 59$   
an corresponds to  $x_1 = 12$ ,  $x_2 = 7$
- stop

## Back to Problem LP3

$$\begin{aligned} \max \quad & 2x_1 + 5x_2 \\ \text{s.t.} \quad & 5x_1 + 2x_2 \leq 81 \\ & 2x_1 + 8x_2 \leq 80 \\ & x_1 \geq 14. \\ & x_1, x_2 \geq 0 \end{aligned}$$



Its solution is  $x_1=14$  and  $x_2=5.5$  with an OV of 55.5.

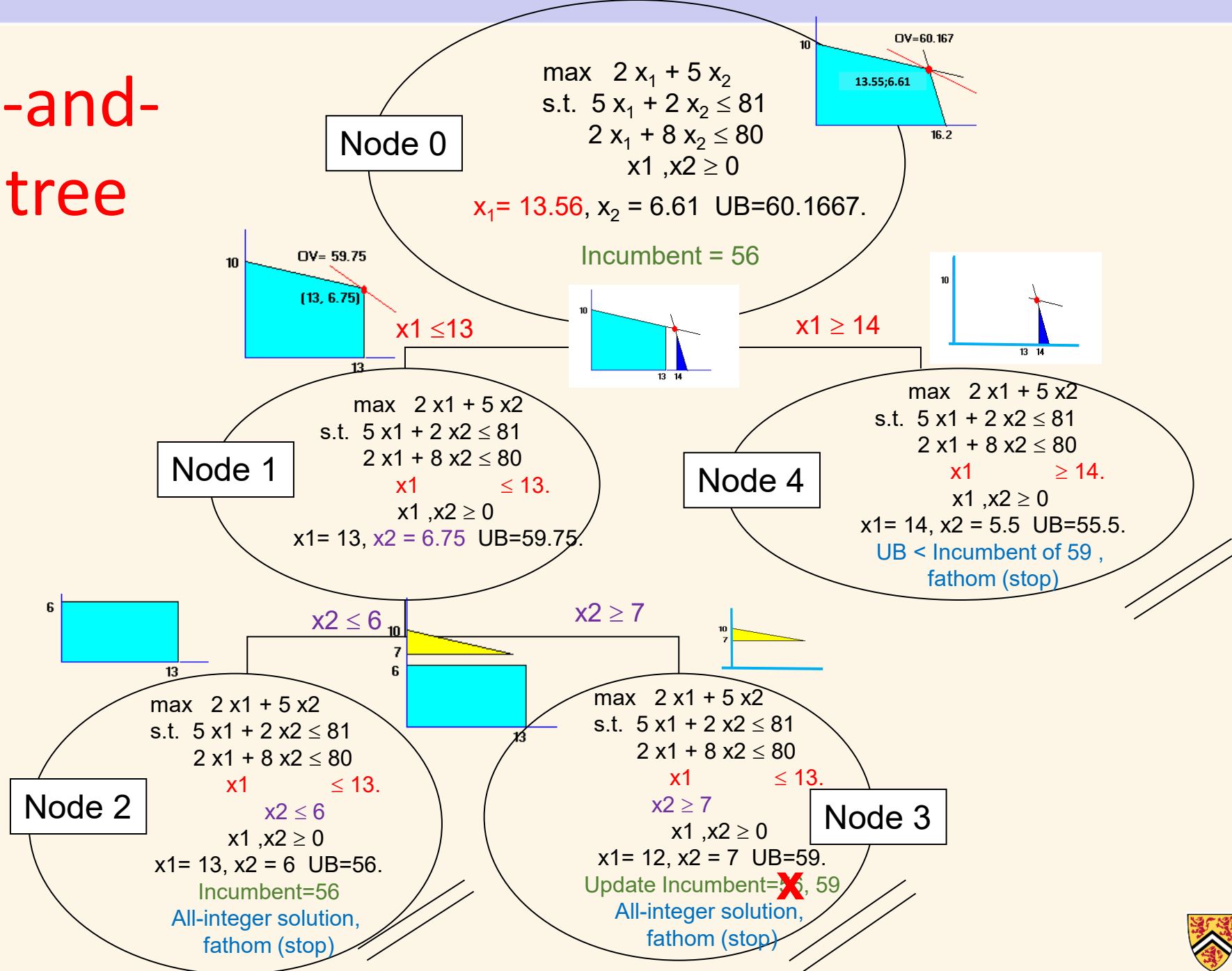
Since its OV is less than the incumbent (=59), this branch is **fathomed**.

Stop. All branches have been explored.

Optimal solution corresponds to incumbent.

So optimal solution is  $x_1 = 12$ ,  $x_2 = 7$  with an objective value (OV) of 59.

# Branch-and-Bound tree



# Terminology

- LP relaxation
- Fathom
- Branching
- Node
- Incumbent

When to stop → When all branches are fathomed

When to fathom a branch → When one of these three conditions occur

1. all-integer solution (for variables that have to be integer)
2. Infeasible LP
3. Current objective is worse than incumbent ( $ov < \text{incumbent}$  for maximization problem)

# The “Classical” branch-and-bound algorithm

1. Start at **node** 0: solve the LP relaxation and get solution  $x^{\text{LP}}$  with objective value  $z^{\text{LP}}$
2. If  $x^{\text{LP}}$  satisfies the integer requirements (all-integer solution), **stop**.
3. If not,
  - a. **Branch** on one of the variables by eliminating the current fractional solution. E.g. if  $x_i^{\text{LP}} = 2.3$ , create two branches with one enforcing  $x_i \leq 2$  and the other  $x_i \geq 3$ .
  - b. Add it to the list of **nodes** to be explored
  - c. (optional) If possible generate a first **incumbent solution** by rounding or other means.
4. As long as there are unexplored **nodes**, do the following
  - a. Choose one of the unexplored nodes
  - b. Solve the corresponding LP
  - c. If all-integer solution is found, update **the incumbent solution** and **fathom** that **node**
  - d. If LP is infeasible, **fathom** that **node**
  - e. If objective  $\leq$  **incumbent objective** (for max. problem), **fathom** that **node**
  - f. Else **Branch**: create two **nodes** and add them to the list of unexplored **nodes**

Once there are no **nodes** to explore, then

- a. If there is an **incumbent solution**, then that is the **optimal solution**.
- b. Else, the **problem is infeasible**.

# Main components of branch-and-bound

$$z = \left\{ \max \sum_{j=1}^m c_j x_j = c^T x : x \in S \right\}$$

## I. Bounding

$$\text{Lower bound} \leq z \leq \text{Upper bound}$$

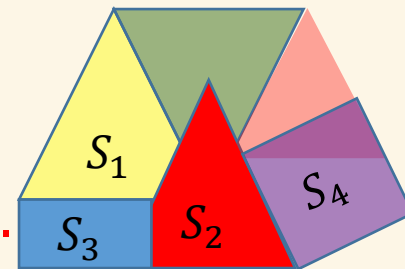
## I. Branching

Let  $S = S_1 \cup S_2 \cup \dots \cup S_k$  be a partition of the set  $S$ , and let

$$z^k = \{\max c^T x : x \in S_k\} \text{ for } k = 1, \dots, K.$$

then

$$z = \max_k z^k.$$

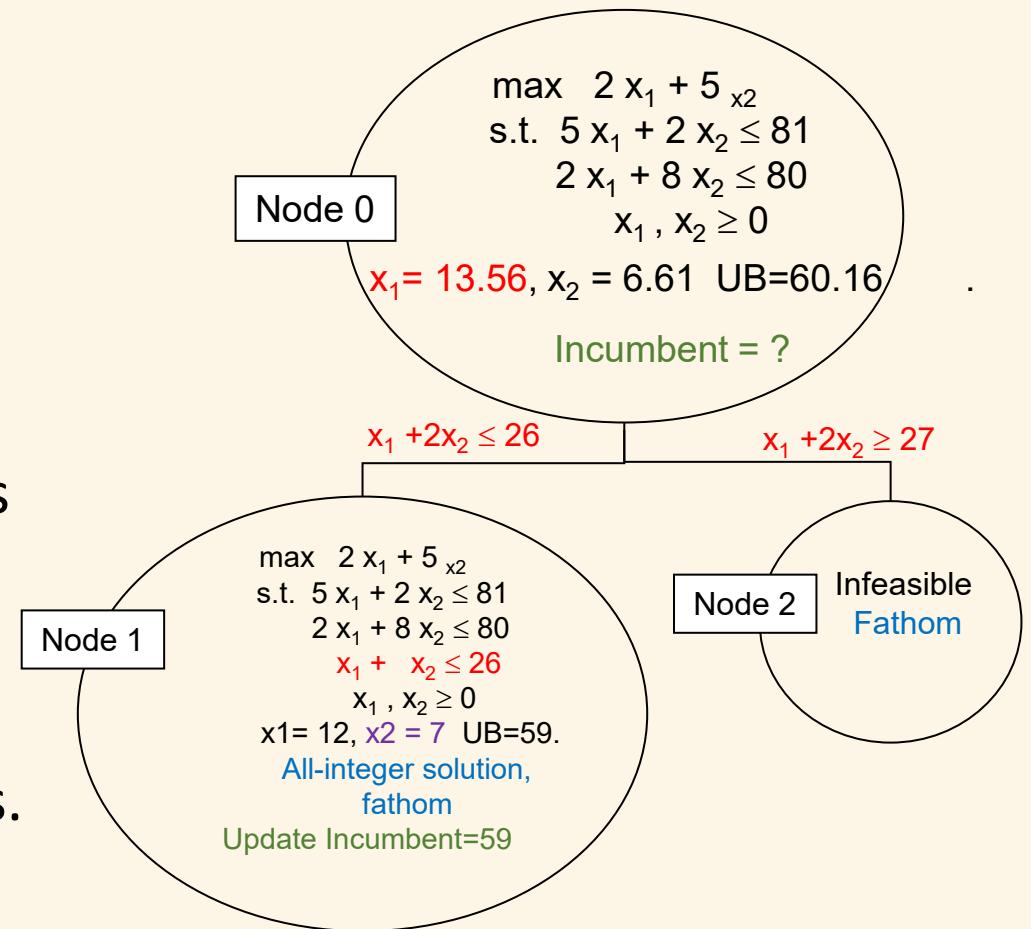


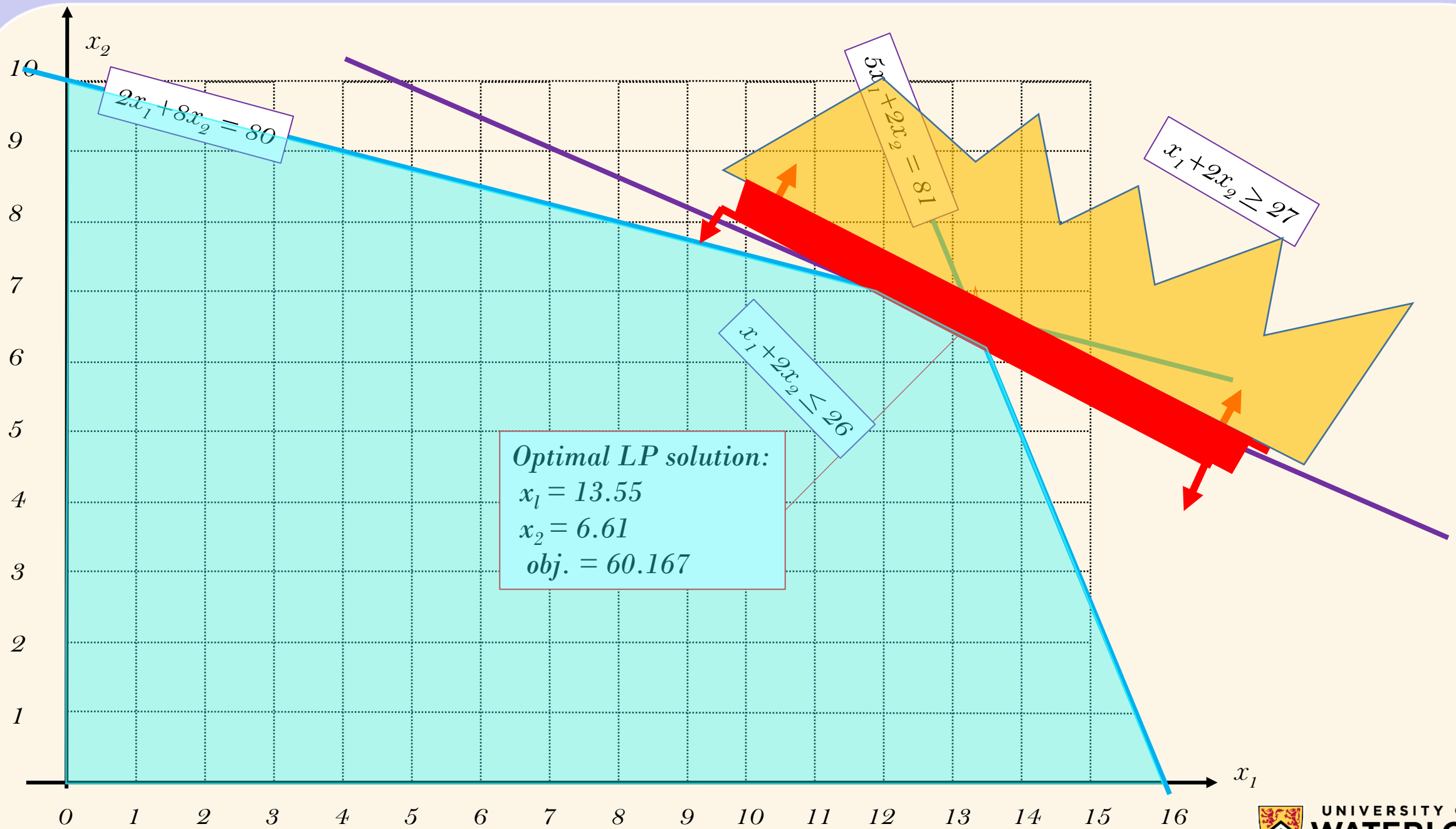
# A different Branching Rule

Not necessary to branch on

$x_1 \leq 13$  versus  $x_1 \geq 14$  or  $x_2 \leq 6$  versus  $x_2 \geq 7$

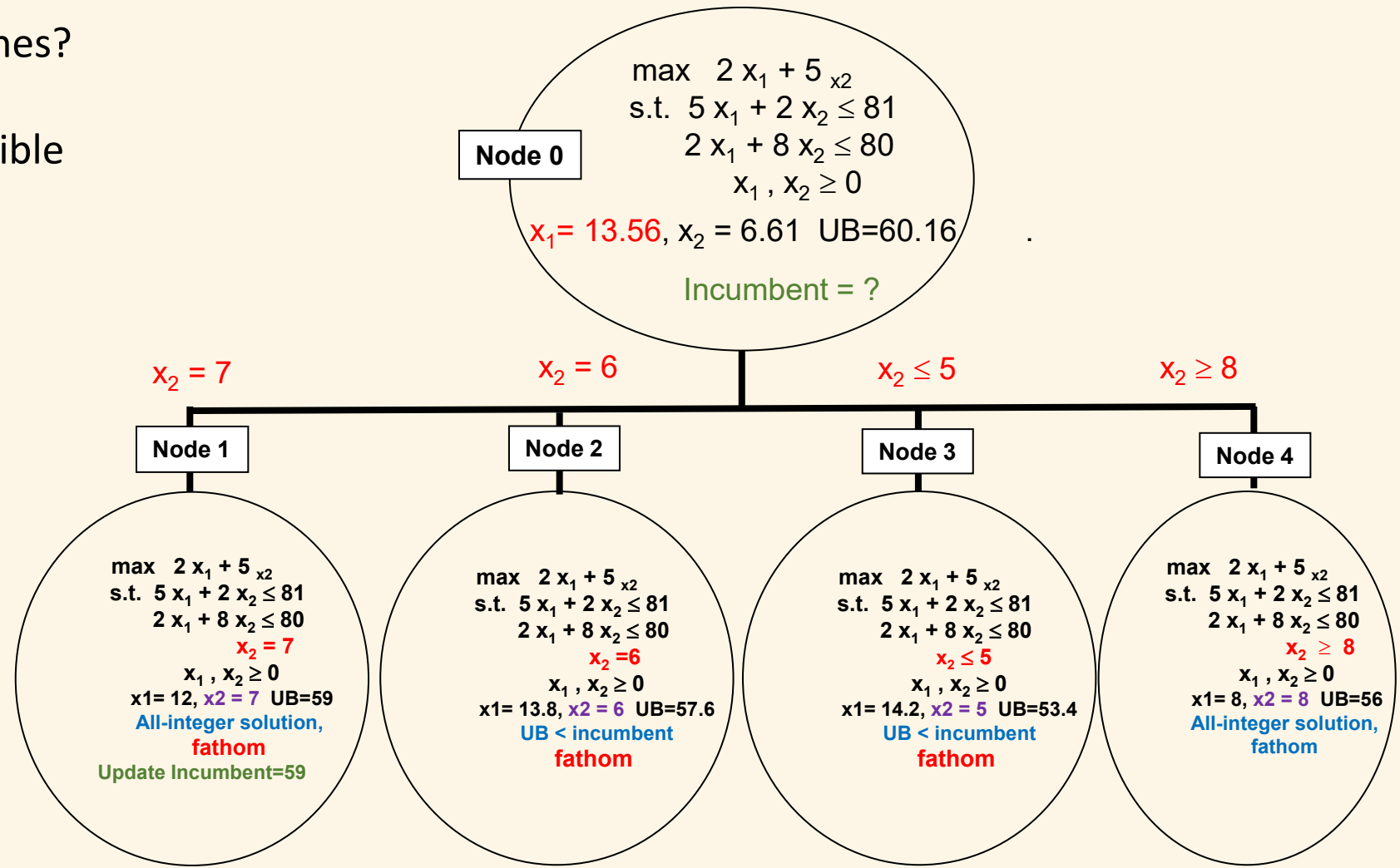
- Could we branch on multiple variables?
- Yes, as long as the union of all branches cover the entire feasible set of integer points
- e.g. how about  
 $x_1 + 2x_2 \leq 26$  versus  $x_1 + 2x_2 \geq 27$
- covers all entire feasible set of integer points.
- At node 0, it is **preferable** to eliminate the LP solution ( $13.56 + 2 \cdot 6.61 = 26.78$ )

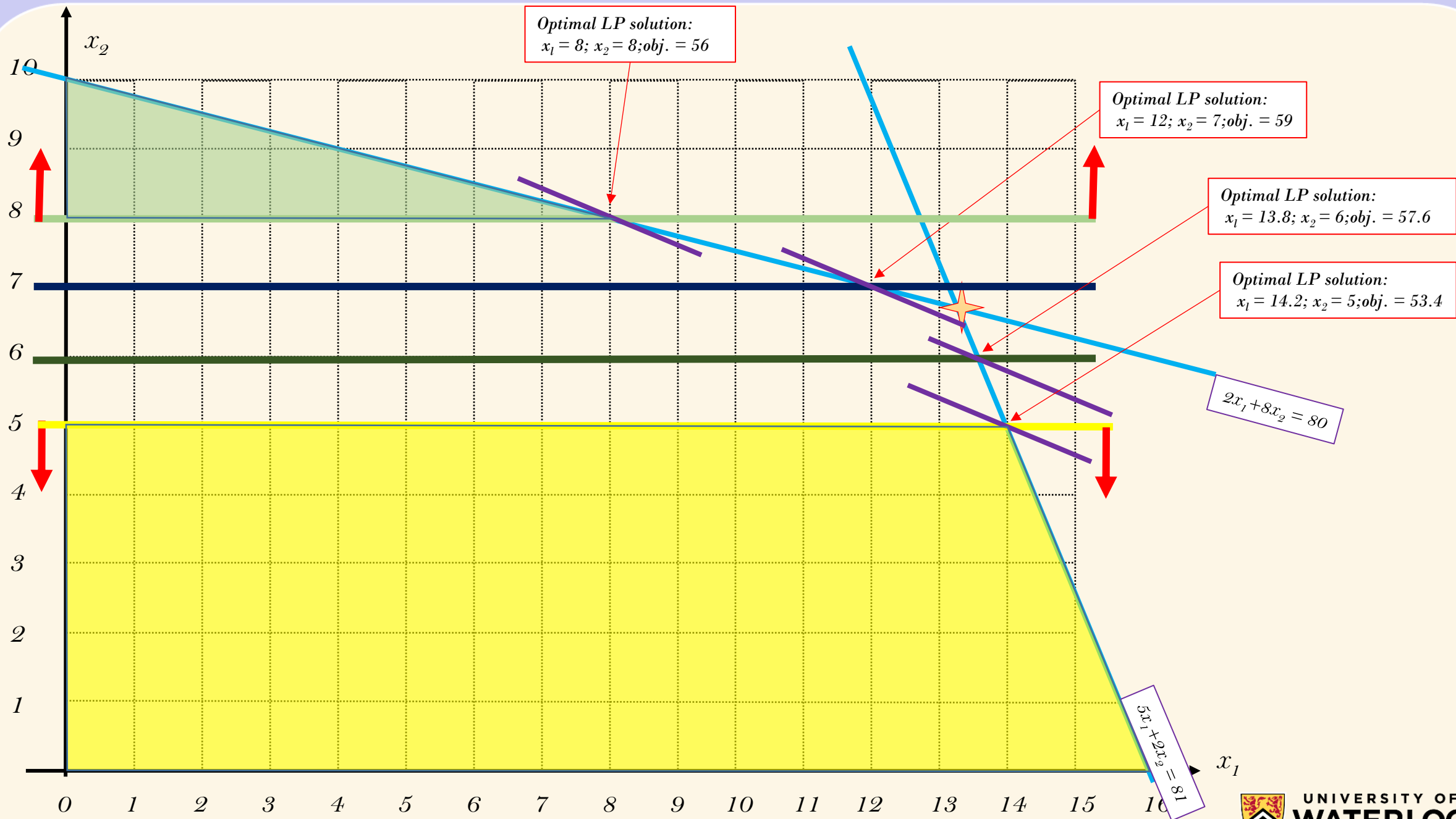




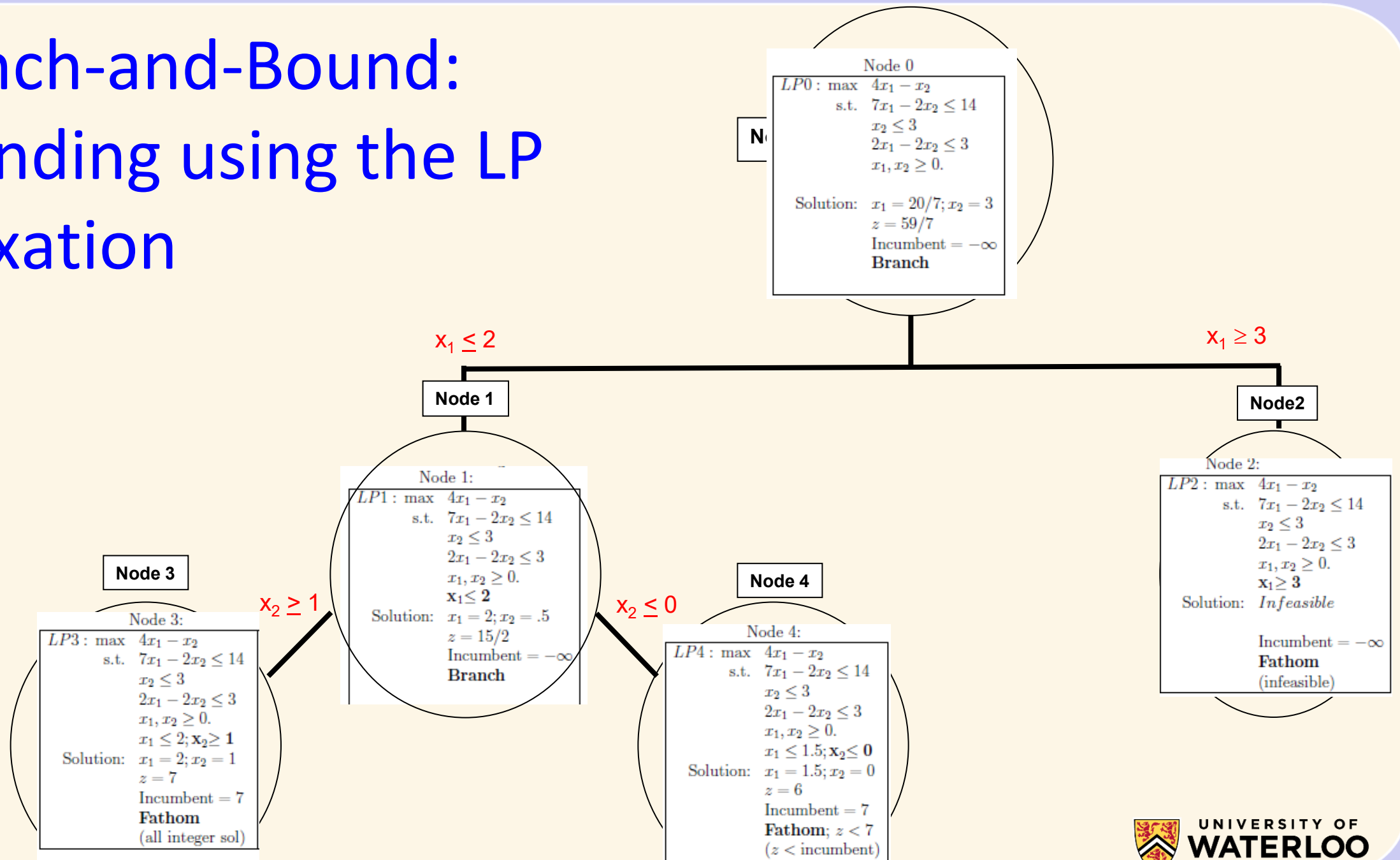
# Another different Branching Rule

- Could we have multiple branches?
- Yes as long as the union of all branches cover the entire feasible set of integer points;
- e.g.
  - $x_2 = 7$
  - $x_2 = 6$
  - $x_2 \leq 5$
  - $x_2 \geq 8$





# Branch-and-Bound: Bounding using the LP relaxation



# Branch-and-Bound: Bounding using a different relaxation

$$\text{Node 0: } \left\{ \begin{array}{ll} \max & 4x_1 - x_2 \\ \text{s.t.} & 7x_1 - 2x_2 \leq 14 \\ & x_2 \leq 3 \\ & 2x_1 - 2x_2 \leq 3 \\ & x_1 \geq 0, \text{ integer;} \\ & x_2 \geq 0; \end{array} \right\}$$

$$x_1 = 2; x_2 = .5, z = 7.5$$

$$\text{Node 1: } \left\{ \begin{array}{ll} \max & 4x_1 - x_2 \\ \text{s.t.} & 7x_1 - 2x_2 \leq 14 \\ & x_2 \leq 3 \\ & 2x_1 - 2x_2 \leq 3 \\ & x_1 > 0, \text{ integer;} \\ & x_2 \leq 0 \\ & x_2 \leq 0 \end{array} \right\}$$

$$x_1 = 1; x_2 = 0, z = 4$$

$$x_2 \leq 0$$

$$x_2 \geq 1$$

$$\text{Node 2: } \left\{ \begin{array}{ll} \max & 4x_1 - x_2 \\ \text{s.t.} & 7x_1 - 2x_2 \leq 14 \\ & x_2 \leq 3 \\ & 2x_1 - 2x_2 \leq 3 \\ & x_1 \geq 0, \text{ integer;} \\ & x_2 \geq 0; \\ & x_2 \geq 1 \end{array} \right\}$$

$$x_1 = 2; x_2 = 1, z = 7$$



# The “General” branch-and-bound algorithm

1. Start at **node 0**:
2. Get a **bound** (**Upper Bound** for max. problem) and
  1. (optional) If possible generate a first **incumbent solution** by rounding or using heuristics.
3. If **incumbent** = **upper bound**; **stop**.
4. Else devise a **branching rule** and
  - a. create multiple problems, one for each node
  - b. Add them to the list of **nodes** to be explored
4. As long as there are unexplored **nodes**, do the following
  - a. Choose one problem (node)
  - b. Find a bound for it
  - c. If **upper bound**  $\leq$  **incumbent objective** (for max. problem), **fathom** that **node**
  - d. If all-integer solution with a better objective is found, update **the incumbent solution**
  - e. Else **Branch**: create more **nodes** and add them to the list of unexplored **nodes**

Once there no **nodes** to explore, then

- a. If there is an **incumbent solution**, then that is the **optimal solution**.
- b. Else, the **problem is infeasible**.

# Choices in Branch-and-Bound

- The branching rule
- The bounding method
- The order in which nodes are processed
- Enhancements
  - Heuristics
  - Constraints/valid cuts

# Branch-and-bound for Mixed-Integer Programs

## Summary:

- Branch-and-bound method
  - branching
  - bounding

Next → Cutting plane methods and branch-and-cut