OMIS 6000

Week 6:

- The integer lattice and the branch-and-bound algorithm
- Scheduling, covering, and assignment problems
- Logical constraints (e.g., conjunctions, disjunctions)

Adam Diamant, Schulich School of Business, Winter 2025

Integer Programs

Divisibility

- Linear programming (LP) allows us to solve large-scale problems. It gives answers in terms of continuous variables (e.g., rational numbers).

Indivisibility

- There are many situations where we need solutions to problems which are not allowed to fall in a continuous range (e.g., Integer/Binary).

Integrality Conditions

MAX: $Z = 350X_1 + 300X_2$ } profit

s.t. $X_1 + X_2 \leq 200$ } filters

$9X_1 + 6X_2 \leq 1566$ } labour

$12X_1 + 16X_2 \leq 2880$ } bolts

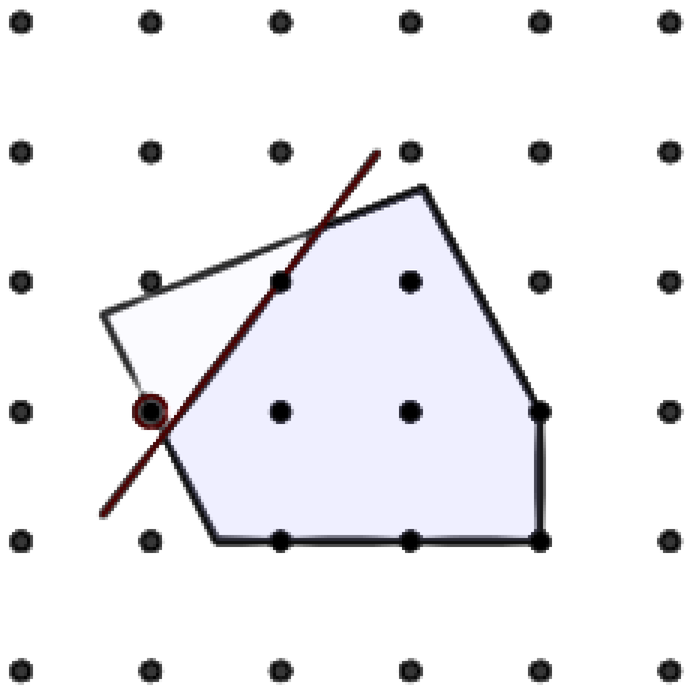
$X_1, X_2 \geq 0$ } non-negativity

X_1, X_2 must be integers } integrality

Integrality conditions are easily stated but make the problem difficult to solve.

Integer vs. Linear Programs

Integrality significantly increases the computational complexity of the problem.



Integer vs. Linear Programs

1. Formulate the IP problem.
2. Solve the **LP relaxation**.
3. If you get integer solutions, stop.
   – You are lucky. You would get the same solution if you had originally solved the IP.
4. If you do not get integer solutions, the LP relaxation provides a **bound** for the IP.
   - For ***maximization*** problems, the LP relaxation is an upper bound to the optimal solution. For ***minimization*** problems, the LP relaxation is a lower bound to the optimal solution.

Integer Programming (IP)

Rounding Example

Machine Shop Example

Why can't we just round?

- What about if we did this?
  1. Formulate an IP problem.
  2. Solve the **LP relaxation**.
  3. Round the optimal solutions to the nearest integer (you can also choose to round all answers down or round all answers up).

- Why does this not work in general?
  – The rounded solution may be **infeasible**.
  – The rounded solution may be **suboptimal**.

Rounding Example

$$\text{Maximize } Z = 3x_1 + 2x_2$$

$$\begin{aligned} \text{subject to:} \quad 3x_1 + x_2 &\leq 9 \\ x_1 + 3x_2 &\leq 7 \\ -x_1 + x_2 &\leq 1 \end{aligned}$$

$$x_1, x_2 \geq 0 \text{ and integer}$$

The optimal solution is $(x_1, x_2) = (3,0)$ with OFV = 9.

Rounding Example

$$\text{Maximize } Z = 3x_1 + 2x_2$$

$$\begin{aligned} \text{subject to: } \quad 3x_1 + x_2 &\leq 9 \\ x_1 + 3x_2 &\leq 7 \\ -x_1 + x_2 &\leq 1 \end{aligned}$$

$$x_1, x_2 \geq 0 \ \cancel{\text{and integer}}$$

The optimal solution is $(x_1, x_2) = (2.5, 1.5)$ with OFV = 10.5.

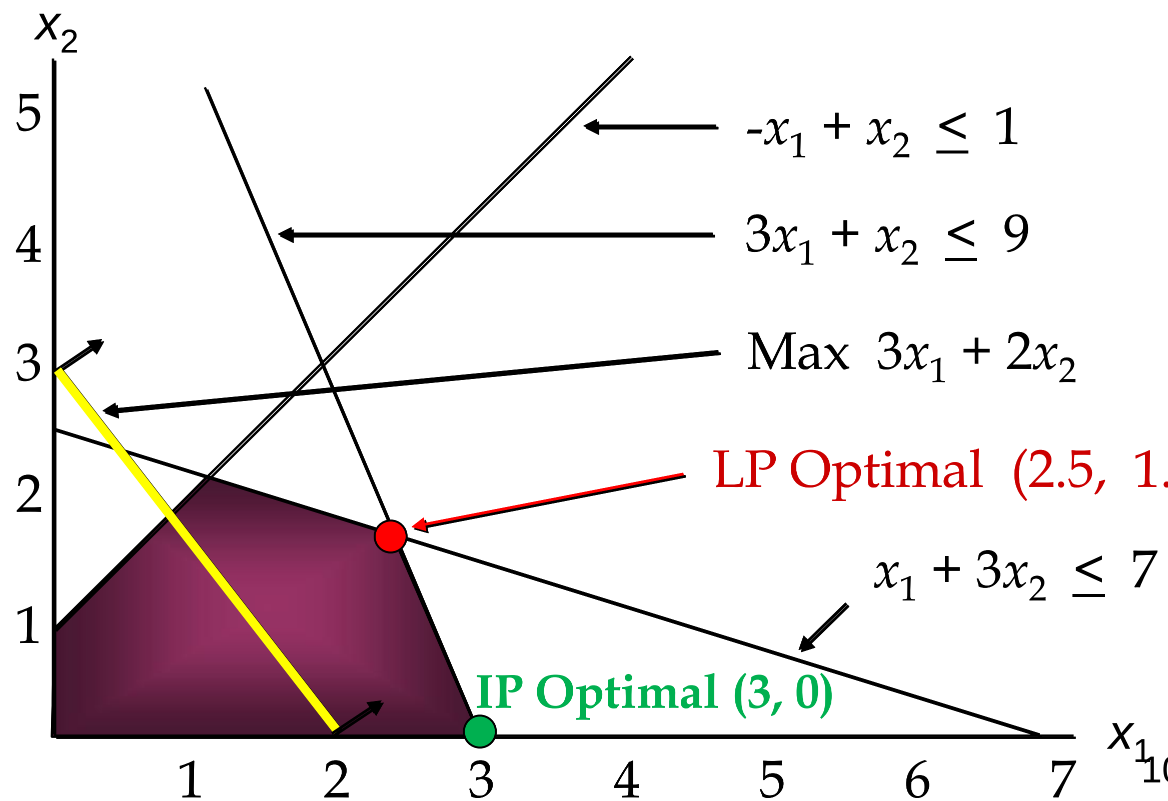
Rounding Example

If we solve the problem as an LP and **ignore the integer constraints**, the optimal solution gives fractional values for both x_1 and x_2. From the graph, the optimal solution is:

Optimal LP Solution:

(x_1, x_2) = (2.5,1.5)

***OFV* = 10.5**



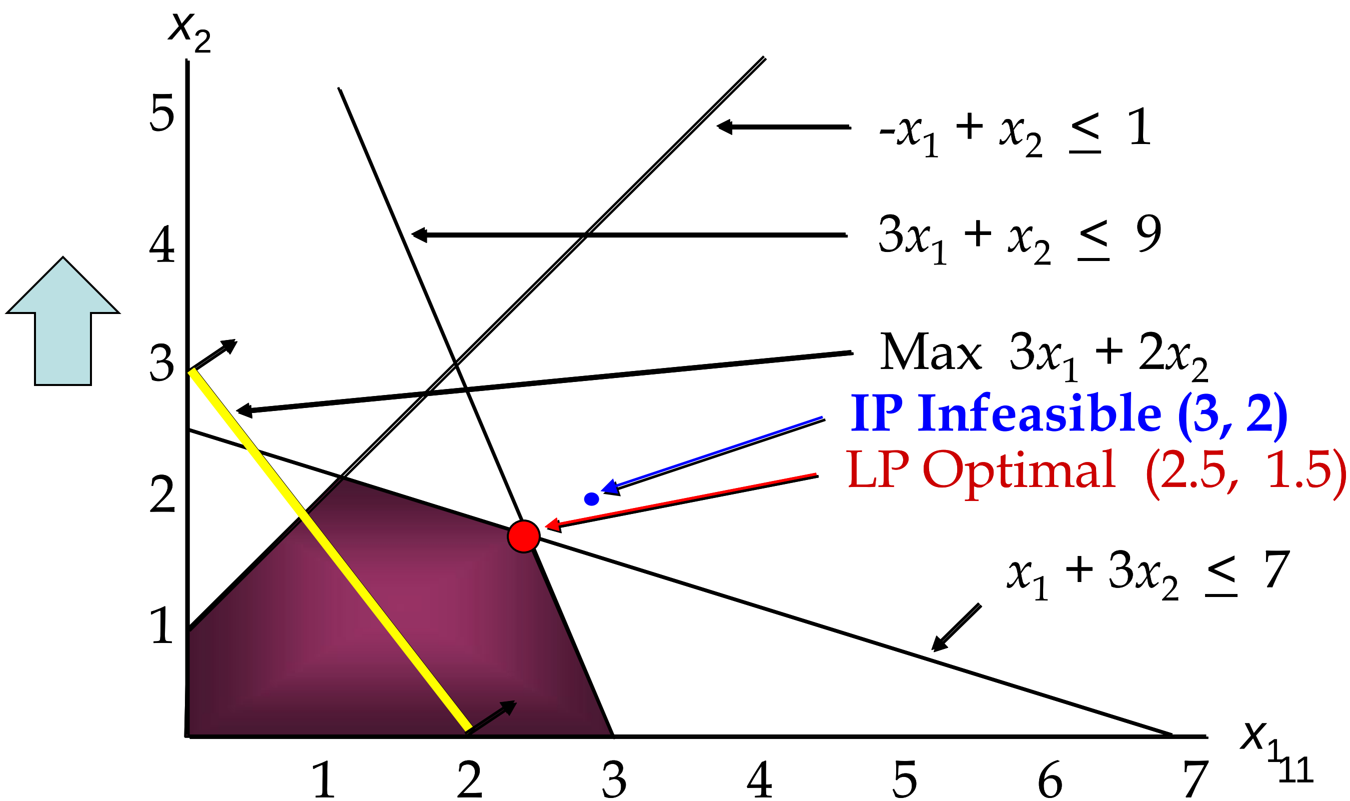
Rounding Example

If we round the fractional solution (2.5, 1.5) from the LP problem, we get (3, 2). This point lies outside the feasible region making it an **infeasible solution**.

Rounded-Up LP Solution:

(x_1, x_2) = (3, 2)

***OFV* = 13**



Rounding Example

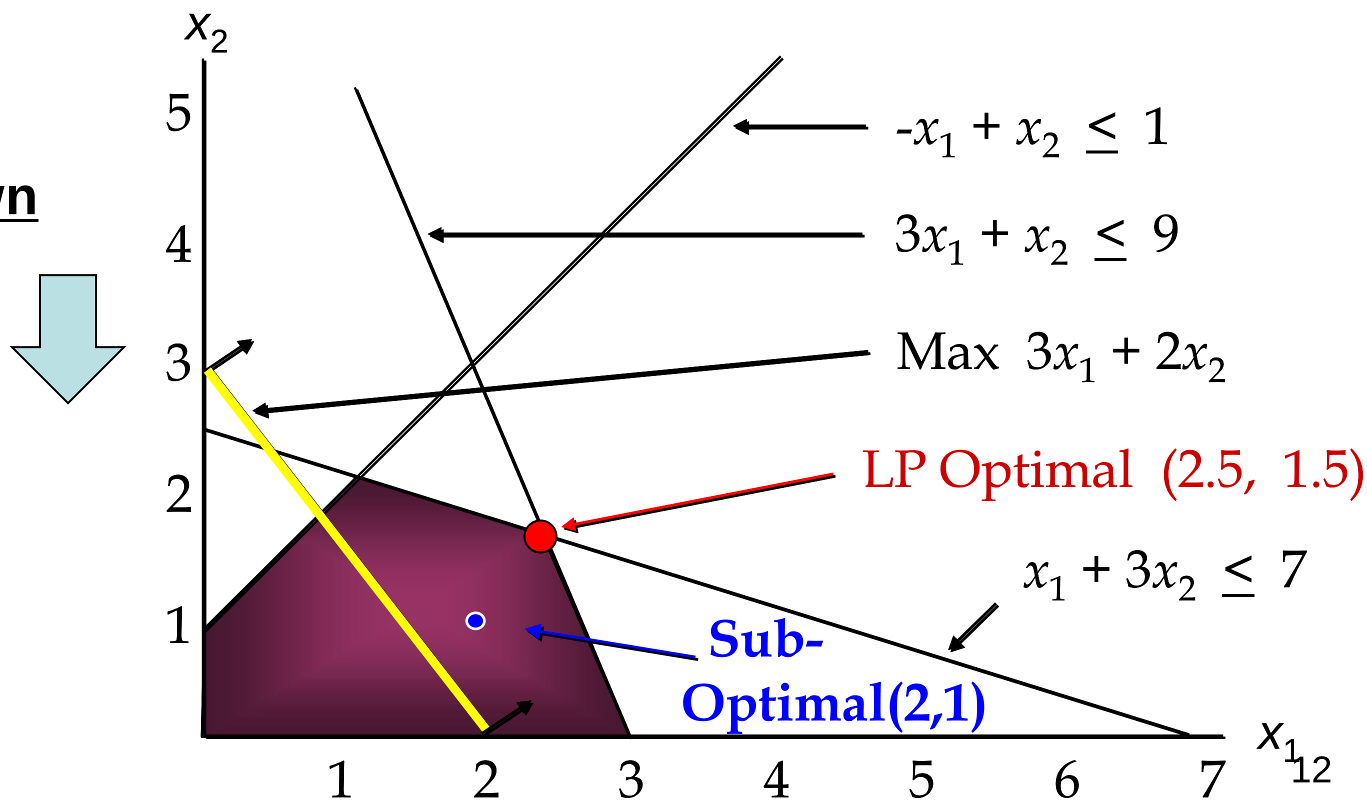
By rounding the optimal LP solution down to (2, 1), the solution is within the feasible region. Substituting the values into the objective function gives *OFV* = 8 (**suboptimal**).

<u>Rounded-Down</u>

<u>LP Solution:</u>

(x_1, x_2) = (2,1)

***OFV* = 8**



Solving Integer Programs

The *Branch-and-Bound* Technique

- Requires the solution of a series of LP problems which are called *candidate problems*.
- **Strengths:** This technique can solve any IP.
- **Weaknesses:** The runtime of the best solution procedure is an *exponential* function of the input (e.g., when the decision variables are binary, we may have to enumerate all possible solutions.)
- **Practically:** It may take lots of time to solve, especially for formulations with many decision variables! Often, though, problems solve quickly.

Machine Shop Example

Machine Shop Example

A small machine shop is experiencing strong growth and wants to expand it business. To do so, more presses and lathes need to be purchased and room on the manufacturing floor needs to be allocated to these new machines. Presses require 15ft^2 of floor space while lathes require 30ft^2. The purchase price of presses and lathes are $\$8000$ are $\$4000$ each, while the marginal daily profit associated with using each machine is $\$100$ and $\$150$, respectively. The expansion budget is $\$40{,}000$ and there is only 200ft^2 of empty floor space. How many of each type of machine should the owner buy to maximize profit?

Machine Shop Example

Let x_1 be the number of presses purchased and x_2 be the number of lathes purchased.

$$\textbf{Maximize } Z = 100x_1 + 150x_2$$

$$8000x_1 + 4000x_2 \leq 40000$$

$$15x_1 + 30x_2 \leq 200$$

$$x_1, x_2 \geq 0 \textit{ and Integer}$$

Machine Shop Example

Let x_1 be the number of presses purchased and x_2 be the number of lathes purchased.

$$\textbf{Maximize } Z = 100x_1 + 150x_2$$

$$8000x_1 + 4000x_2 \leq 40000$$

$$15x_1 + 30x_2 \leq 200$$

$$x_1, x_2 \geq 0$$

If we solve the corresponding **LP**, we get:
$x_1 = 2.22$, $x_2 = 5.56$, and $Z = 1055.56$.

Machine Shop Example

Let x_1 be the number of presses purchased and x_2 be the number of lathes purchased.

$$\textbf{Maximize } Z = 100x_1 + 150x_2$$

$$8000x_1 + 4000x_2 \leq 40000$$

$$15x_1 + 30x_2 \leq 200$$

$$x_1, x_2 \geq 0$$

Let’s round the answer down so it remains feasible: $x_1 = 2$, $x_2 = 5$, and $Z = 950$.

Machine Shop Example

The *Branch-and-Bound* Technique

- A decision tree is created. The *root node* represents all feasible solutions. The algorithm explores branches of the tree (subsets of the solution set).
  - The optimal *integer* solution will always be between the upper bound that is provided by the *LP* solution and the lower bound of the rounded-down integer solution.

$$\mathbf{UB = 1055.66}$$
$$(x_1 = 2.22,\ x_2 = 5.56)$$
$$\mathbf{LB = 950}\ (x_1 = 2,\ x_2 = 5)$$

Machine Shop Example

The *Branch-and-Bound* Technique

$\mathbf{UB = 1055.66}$

$\mathbf{(x_1 = 2.22,\ x_2 = 5.56)}$

$\mathbf{LB = 950\ (x_1 = 2,\ x_2 = 5)}$

$\mathbf{x_2 \leq 5}$

$\mathbf{x_2 \geq 6}$

Machine Shop Example

Let x_1 be the number of presses purchased and x_2 be the number of lathes purchased.

$$\textbf{Maximize } Z = 100x_1 + 150x_2$$

$$8000x_1 + 4000x_2 \leq 40000$$

$$15x_1 + 30x_2 \leq 200$$

$$x_2 \leq 5$$

Left Branch

$$x_1, x_2 \geq 0$$

Machine Shop Example

Let x_1 be the number of presses purchased and x_2 be the number of lathes purchased.

$$\textbf{Maximize } Z = 100x_1 + 150x_2$$

$$8000x_1 + 4000x_2 \leq 40000$$

$$15x_1 + 30x_2 \leq 200$$

$$x_2 \geq 6$$

Right Branch

$$x_1, x_2 \geq 0$$

Machine Shop Example

The *Branch-and-Bound* Technique

$\mathbf{UB = 1055.66}$
$(x_1 = 2.22,\ x_2 = 5.56)$
$\mathbf{LB = 950}\ (x_1 = 2,\ x_2 = 5)$

$\mathbf{UB = 1000}$
$(x_1 = 2.5,\ x_2 = 5)$
$\mathbf{LB = 950}\ (x_1 = 2,\ x_2 = 5)$

$\mathbf{UB = 1033.33}$
$(x_1 = 1.33,\ x_2 = 6)$
$\mathbf{LB = 1000}\ (x_1 = 1,\ x_2 = 6)$

Machine Shop Example

The *Branch-and-Bound* Technique

$\mathbf{UB = 1055.66}$
$\mathbf{(x_1 = 2.22,\ x_2 = 5.56)}$
$\mathbf{LB = 950\ (x_1 = 2,\ x_2 = 5)}$

Stop Exploring (For Now…)

$\mathbf{UB = 1000}$
$\mathbf{(x_1 = 2.5,\ x_2 = 5)}$
$\mathbf{LB = 950\ (x_1 = 2,\ x_2 = 5)}$

UB (Left) = 1000 < 1033 = UB (Right)

$\mathbf{UB = 1033.33}$
$\mathbf{(x_1 = 1.33,\ x_2 = 6)}$
$\mathbf{LB = 1000\ (x_1 = 1,\ x_2 = 6)}$

Machine Shop Example

The *Branch-and-Bound* Technique

$\mathbf{UB = 1055.66}$
$\mathbf{(x_1 = 2.22, x_2 = 5.56)}$
$\mathbf{LB = 950\ (x_1 = 2, x_2 = 5)}$

$\mathbf{UB = 1000}$
$\mathbf{(x_1 = 2.5, x_2 = 5)}$
$\mathbf{LB = 950\ (x_1 = 2, x_2 = 5)}$

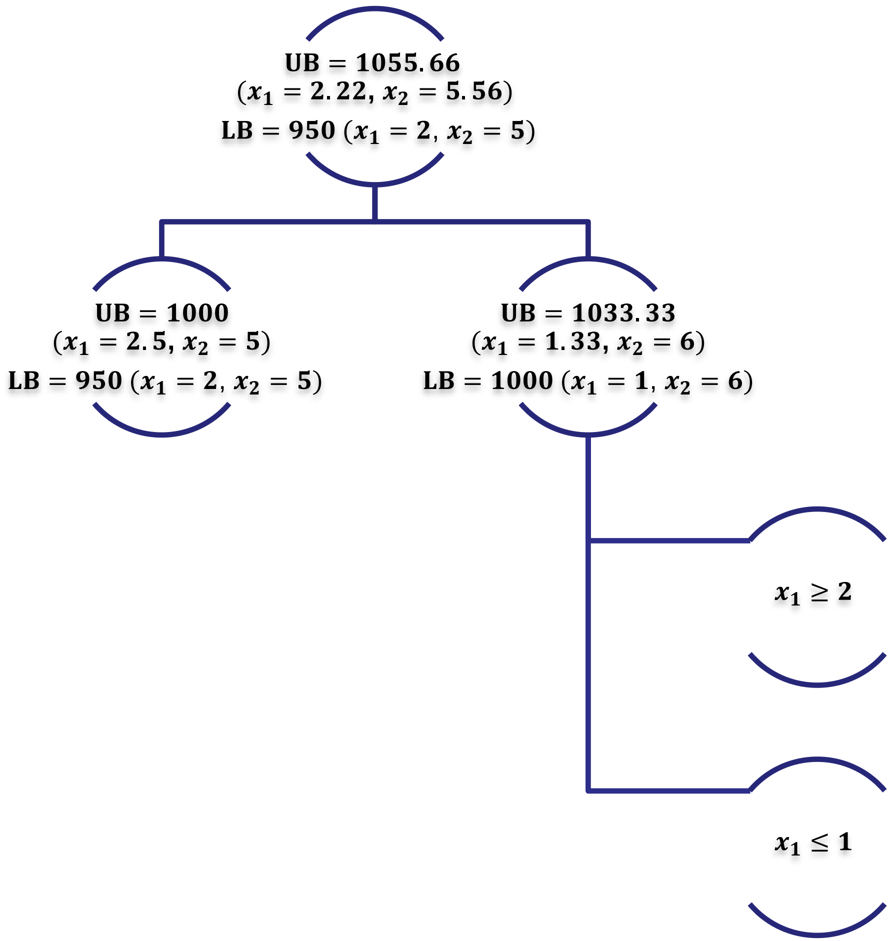
Keep Exploring

$\mathbf{UB = 1033.33}$
$\mathbf{(x_1 = 1.33, x_2 = 6)}$
$\mathbf{LB = 1000\ (x_1 = 1, x_2 = 6)}$

UB (Left) = 1000 < 1033 = UB (Right)

Machine Shop Example

The *Branch-and-Bound* Technique



Machine Shop Example

Let x_1 be the number of presses purchased and x_2 be the number of lathes purchased.

$$\textbf{Maximize } Z = 100x_1 + 150x_2$$

$$8000x_1 + 4000x_2 \leq 40000$$

$$15x_1 + 30x_2 \leq 200$$

$$x_2 \geq 6$$

Left Branch

$$x_1 \leq 1$$

$$x_1, x_2 \geq 0$$

Machine Shop Example

Let x_1 be the number of presses purchased and x_2 be the number of lathes purchased.

$$\textbf{Maximize } Z = 100x_1 + 150x_2$$

$$8000x_1 + 4000x_2 \leq 40000$$

$$15x_1 + 30x_2 \leq 200$$

$$x_2 \geq 6$$

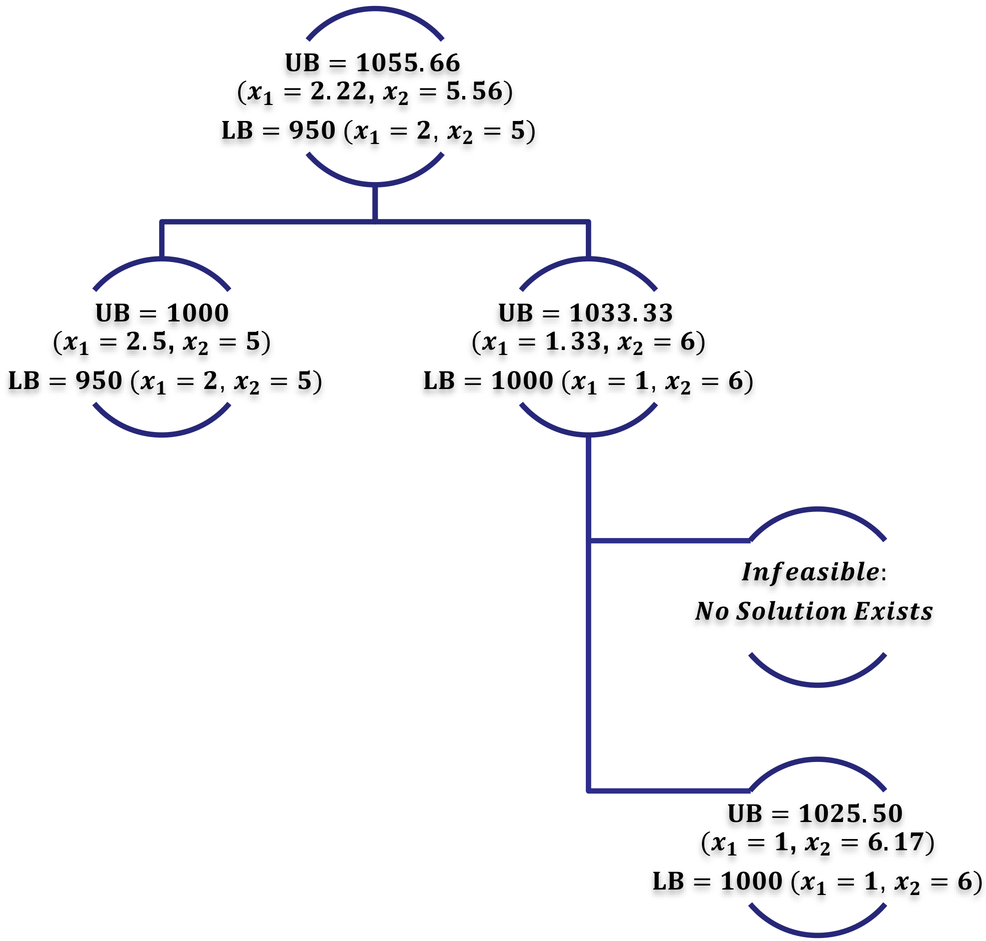
Right Branch

$$x_1 \geq 2$$

$$x_1, x_2 \geq 0$$

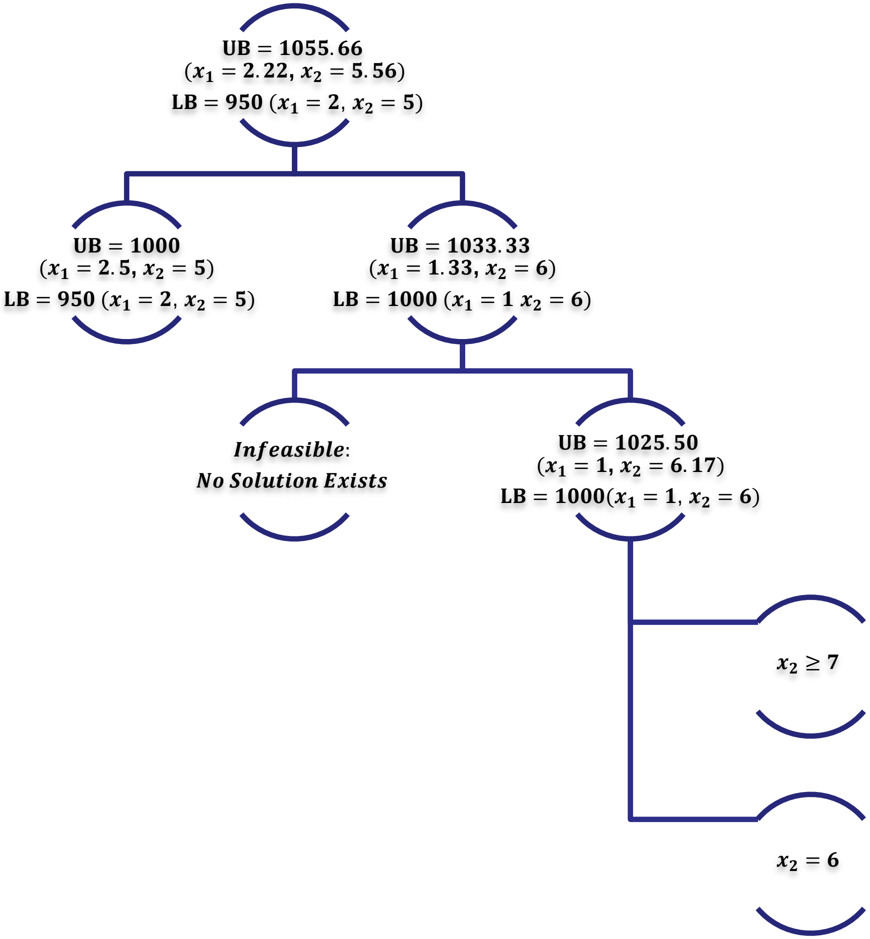
Machine Shop Example

The *Branch-and-Bound* Technique



Machine Shop Example

The *Branch-and-Bound* Technique



Machine Shop Example

Let x_1 be the number of presses purchased and x_2 be the number of lathes purchased.

$$\textbf{Maximize } Z = 100x_1 + 150x_2$$

$$8000x_1 + 4000x_2 \leq 40000$$

$$15x_1 + 30x_2 \leq 200$$

$$x_2 = 6$$

Left Branch

$$x_1 \leq 1$$

$$x_1, x_2 \geq 0$$

Machine Shop Example

Let x_1 be the number of presses purchased and x_2 be the number of lathes purchased.

$$\textbf{Maximize } Z = 100x_1 + 150x_2$$

$$8000x_1 + 4000x_2 \leq 40000$$

$$15x_1 + 30x_2 \leq 200$$

$$x_2 \geq 7$$

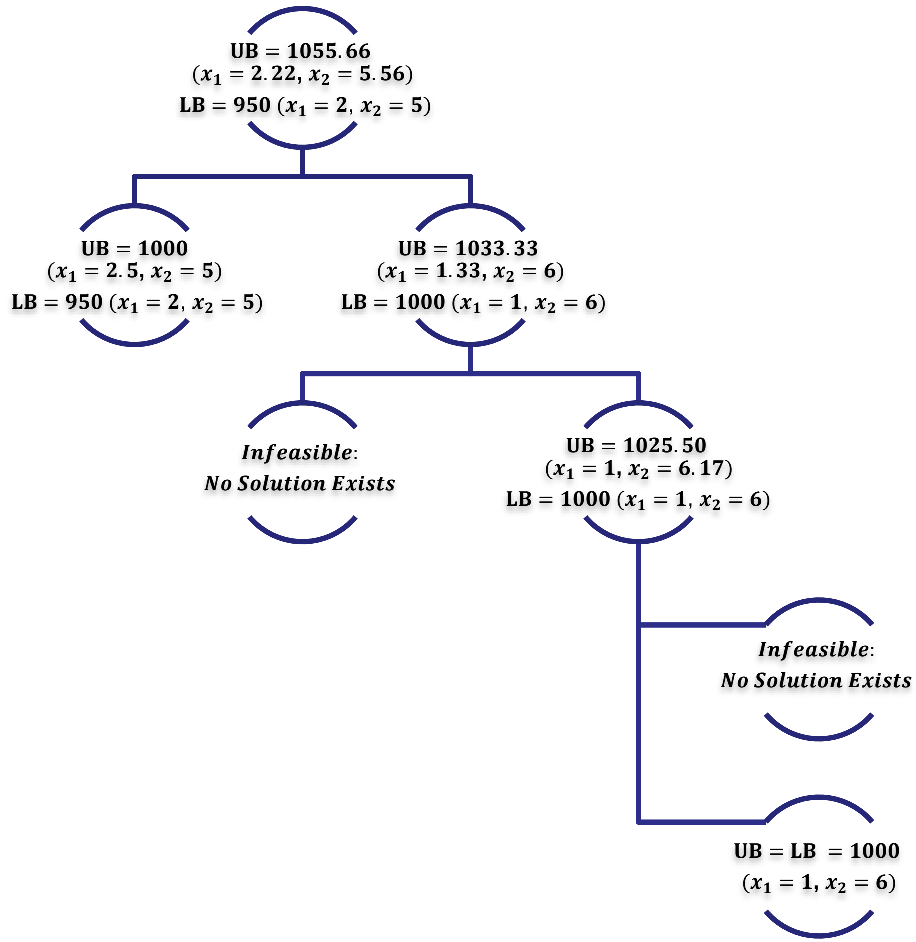
Right Branch

$$x_1 \geq 2$$

$$x_1, x_2 \geq 0$$

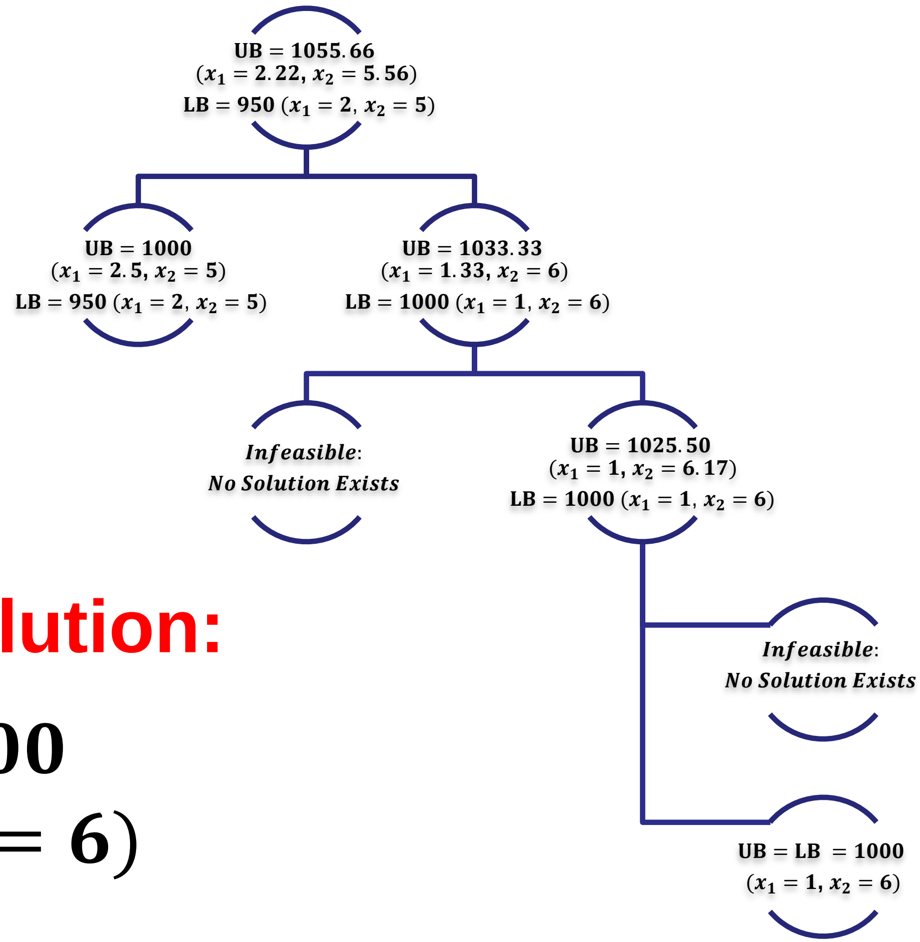
Machine Shop Example

The *Branch-and-Bound* Technique



Machine Shop Example

The *Branch-and-Bound* Technique



Optimal Solution:

$$Z = 1000$$
$$(x_1 = 1,\ x_2 = 6)$$

Branch and Bound

The algorithm solves *multiple* LPs!

- Systematically explores candidate solutions by representing the problem as a decision tree.
- Each iteration explores branches of this tree, which represent subsets of the solution set.
- As we add constraints, the models get larger.
- Each model provides an upper bound (**LP solution**) and a lower bound (**rounded solution**).
- Stop iterating when all parts of the solution space have been explored and the current solution that has been found represents the best possible value.

Branch-and-Bound

At each iteration k of the algorithm, we compute an upper bound $z_{ub}^{(k)}$ and a lower bound $z_{lb}^{(k)}$.

- We also add new constraints such that for some variable j, one branch of the tree solves the problem where $x_j \leq \left\lfloor x_j^{(k)} \right\rfloor$ and $x_j \geq \left\lceil x_j^{(k)} \right\rceil$.

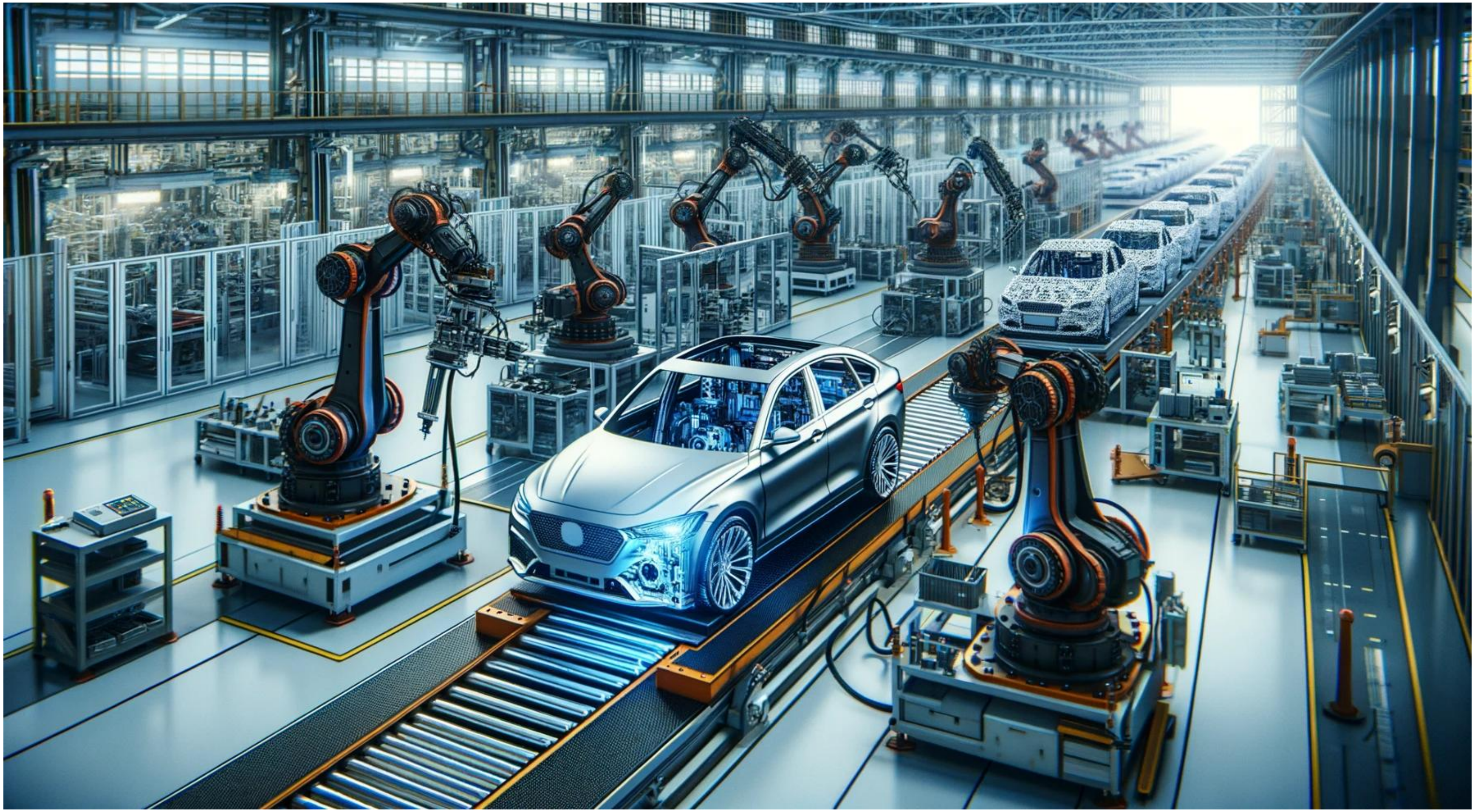
- A decision tree is created. The *root node* represents all feasible solutions. The algorithm creates and then explores branches of the tree.
  - If the optimal solution value to the **LP relaxation** is smaller than the current lower bound, we do not need to consider the sub-problem further (prune the tree).

Branch-and-Bound

Theorem: The Branch-and-Bound algorithm, used to solve an integer program as a series of linear programming relaxations, converges to the optimal solution in a finite number of iterations.

Proof: In-Class.

Assembly Line Scheduling



Assembly Line Scheduling

- How do you assign employees to shifts to minimize wage costs? Remember, all scheduling constraints must be adhered to!
- Constraints could be imposed by the company or an external body (e.g., union).
- You ***cannot*** assign half a worker to a shift.
- In many applications (e.g., nurses, customer service, manufacturing), multiple employees must work together during the same shift.

Assembly Line Scheduling

Day of Week	Workers Needed
Sunday	18
Monday	27
Tuesday	22
Wednesday	26
Thursday	25
Friday	21
Saturday	19

Shift	Days Off	Wage
1	Sun & Mon	$680
2	Mon & Tue	$705
3	Tue & Wed	$705
4	Wed & Thr	$705
5	Thr & Fri	$705
6	Fri & Sat	$680
7	Sat & Sun	$655

Determine how many workers should be assigned to each shift so that the number of workers needed on each day is satisfied. Do this in such a way as to **minimize costs**!

Assembly Line Scheduling

Define the objective

Minimize the total wages

Define the decision variables

Assembly Line Scheduling

Define the objective

Minimize the total wages

Define the decision variables

$x_i =$ **the number of workers assigned to shift *i* where *i* $= 1, .., 7$**

Assembly Line Scheduling

Write the mathematical objective function

Minimize $Z =$

Shift	Days Off	Wage
1	Sun & Mon	$680
2	Mon & Tue	$705
3	Tue & Wed	$705
4	Wed & Thr	$705
5	Thr & Fri	$705
6	Fri & Sat	$680
7	Sat & Sun	$655

Assembly Line Scheduling

Write the mathematical objective function

$$\text{Minimize } Z = 680x_1 + 705x_2 + 705x_3 + 705x_4 + 705x_5 + 680x_6 + 655x_7$$

Shift	Days Off	Wage
1	Sun & Mon	$680
2	Mon & Tue	$705
3	Tue & Wed	$705
4	Wed & Thr	$705
5	Thr & Fri	$705
6	Fri & Sat	$680
7	Sat & Sun	$655

Assembly Line Scheduling

Formulate the constraints

	Mon	Tue	Wed	Thu	Fri	Sat	Sun	# of workers assigned to each shift
Shift 1								x_1
Shift 2								x_2
Shift 3								x_3
Shift 4								x_4
Shift 5								x_5
Shift 6								x_6
Shift 7								x_7
# of workers needed for each day	27	22	26	25	21	19	18	

Workers required each day:

Assembly Line Scheduling

Formulate the constraints

	Mon	Tue	Wed	Thu	Fri	Sat	Sun	# of workers assigned to each shift
Shift 1								x_1
Shift 2								x_2
Shift 3								x_3
Shift 4								x_4
Shift 5								x_5
Shift 6								x_6
Shift 7								x_7
# of workers needed for each day	27	22	26	25	21	19	18	

Workers required each day:

$\boldsymbol{x_3 + x_4 + x_5 + x_6 + x_7 \geq 27}$ (Monday Constraint)

$\boldsymbol{x_4 + x_5 + x_6 + x_7 + x_1 \geq 22}$ (Tuesday Constraint)

$\boldsymbol{x_5 + x_6 + x_7 + x_1 + x_2 \geq 26}$ (Wednesday Constraint)

$\boldsymbol{x_6 + x_7 + x_1 + x_2 + x_3 \geq 25}$ (Thursday Constraint)

$\boldsymbol{x_7 + x_1 + x_2 + x_3 + x_4 \geq 21}$ (Friday Constraint)

$\boldsymbol{x_1 + x_2 + x_3 + x_4 + x_5 \geq 19}$ (Saturday Constraint)

$\boldsymbol{x_2 + x_3 + x_4 + x_5 + x_6 \geq 18}$ (Sunday Constraint)

Assembly Line Scheduling

Minimize $Z = 680x_1 + 705x_2 + 705x_3 + 705x_4 + 705x_5 + 680x_6 + 655x_7$

Subject to:

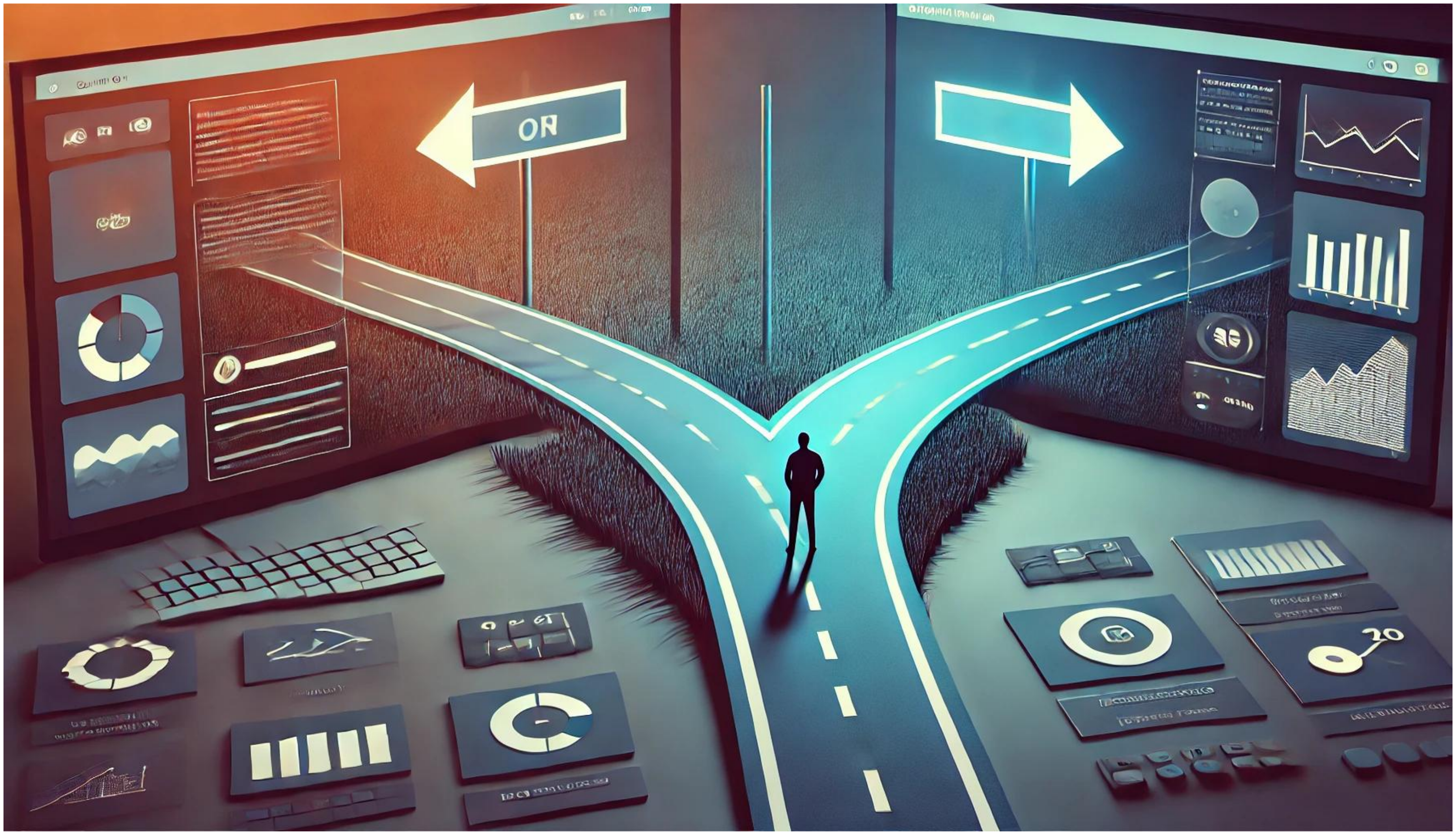
$x_3 + x_4 + x_5 + x_6 + x_7 \geq 27$	(Monday Constraint)
$x_4 + x_5 + x_6 + x_7 + x_1 \geq 22$	(Tuesday Constraint)
$x_5 + x_6 + x_7 + x_1 + x_2 \geq 26$	(Wednesday Constraint)
$x_6 + x_7 + x_1 + x_2 + x_3 \geq 25$	(Thursday Constraint)
$x_7 + x_1 + x_2 + x_3 + x_4 \geq 21$	(Friday Constraint)
$x_1 + x_2 + x_3 + x_4 + x_5 \geq 19$	(Saturday Constraint)
$x_2 + x_3 + x_4 + x_5 + x_6 \geq 18$	(Sunday Constraint)
$x_1 \geq 0, \ldots, x_7 \geq 0 \;\; and\ Integer$	(Integrality Constraints)

Assembly Line Scheduling: Python Solution

- Remember to add parameters that ensure the decision variables are integral.
- If you don't do this, you will get a fractional solution, which does not make sense.
- Notice that the optimal solution exceeds the RHS value of some constraints. **Why**?

What managerial intuition do you get from the Python solution?

Binary Programs (BIP)



Binary Programs (BIP)

- Binary variables are integers that assume two values: **0 or 1*.*** They represent ***True or False, Good or Bad, Up or Down, Happy or Sad.***
- These variables can be useful in many practical modeling situations when there is choice….
- Binary programs are especially important when, for each item in the problem, we are concerned with answering the following question:

Should this item be included in an optimal solution?

Yes (1) or No (0).

Logical Constraints

- In many cases, binary variables represent **economic indivisibilities**. For example, a choice is either selected or it is not. There is no selecting a fraction (e.g., a half) of a choice. However, binary constraints are also useful because they can impose constraints that describe logical conditions.

- Logical conditions are identified by the words: **if**, **and**, **or**, **not, nor, both, neither, all, exactly, none, at most, at least, when...**

Logical Constraints

Examples:

- A year-end bonus will be awarded only if the sales target has been achieved.
- A rebate will only be given if the order size exceeds a specific threshold.
- No two retail stores can be opened at the exact same location (one store or the other can be opened at that location, not both).
- A student can only enroll in a course when all the prerequisites are satisfied, and they have completed a total of 36 credit hours.

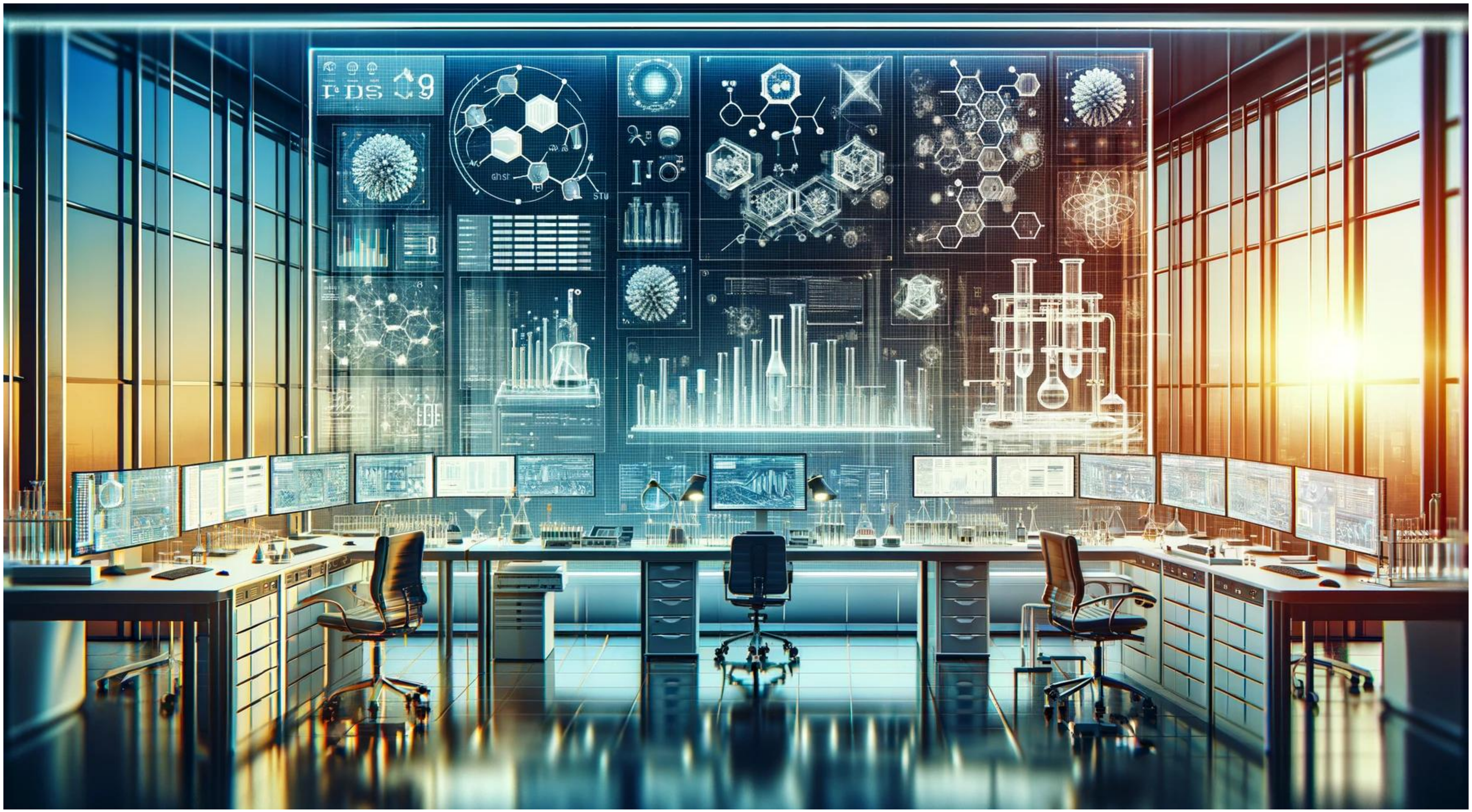
Binary Programs (BIPs)

Project Investment

Covering Problems

Personnel Selection

Project Investment



Project Investment

Amazon® is considering investing in six large R&D projects. The cash required for each investment and the expected net present value (NPV) is given in the table below. The total amount that can be invested is $14 billion. Amazon® wants to maximize its expected NPV; what is the optimal strategy?

***Note:* A project can be selected or left out. One cannot select a fraction of a project because it will surely fail.**

Project	1	2	3	4	5	6
Cash Required ($billions)	$5	$7	$4	$3	$4	$6
NPV added ($billions)	$16	$22	$12	$8	$11	$19

Project Investment

Define the objective

Maximize the expected NPV

Define the decision variables

Write the mathematical objective function

Formulate the resource constraint

Project Investment

Define the objective

Maximize the expected NPV

Define the decision variables

$$x_i = \begin{cases} 1, & if\ we\ invest\ in\ project\ i\ where\ i = 1, \ldots, 6 \\ 0, & \quad otherwise \end{cases}$$

Write the mathematical objective function

Formulate the resource constraint

Project Investment

Define the objective

Maximize the expected NPV

Define the decision variables

$$x_i = \begin{cases} 1, & if\ we\ invest\ in\ project\ i\ where\ i = 1, \dots, 6 \\ 0, & otherwise \end{cases}$$

Write the mathematical objective function

Maximize $Z = 16x_1 + 22x_2 + 12x_3 + 8x_4 + 11x_5 + 19x_6$

Formulate the resource constraint

Project Investment

Define the objective

Maximize the expected NPV

Define the decision variables

$$x_i = \begin{cases} 1, & if\ we\ invest\ in\ project\ i\ where\ i = 1, \ldots, 6 \\ 0, & otherwise \end{cases}$$

Write the mathematical objective function

Maximize $Z = 16x_1 + 22x_2 + 12x_3 + 8x_4 + 11x_5 + 19x_6$

Formulate the resource constraint

$$5x_1 + 7x_2 + 4x_3 + 3x_4 + 4x_5 + 6x_6 \leq 14$$
$$x_i = \{0, 1\} \quad for\ i = 1, .., 6$$

Project Investment

1) Exactly 3 of the 6 projects should be selected.

2) If project 2 is selected, then project 1 must also be selected.

3) Project 1 and project 3 cannot both be selected.

4) Either project 4 is selected or 5 is selected, but not both.

5) Project 3 must be selected if project 1 and 2 are selected.

6) Project 6 must be selected if project 4 and 5 are not selected.

Project Investment

1) Exactly 3 of the 6 projects should be selected.

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 3$$

2) If project 2 is selected, then project 1 must also be selected.

3) Project 1 and project 3 cannot both be selected.

4) Either project 4 is selected or 5 is selected, but not both.

5) Project 3 must be selected if project 1 and 2 are selected.

6) Project 6 must be selected if project 4 and 5 are not selected.

Project Investment

1) Exactly 3 of the 6 projects should be selected.

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 3$$

2) If project 2 is selected, then project 1 must also be selected.

$$x_2 \leq x_1$$

3) Project 1 and project 3 cannot both be selected.

4) Either project 4 is selected or 5 is selected, but not both.

5) Project 3 must be selected if project 1 and 2 are selected.

6) Project 6 must be selected if project 4 and 5 are not selected.

Project Investment

1) Exactly 3 of the 6 projects should be selected.

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 3$$

2) If project 2 is selected, then project 1 must also be selected.

$$x_2 \leq x_1$$

3) Project 1 and project 3 cannot both be selected.

$$x_1 + x_3 \leq 1$$

4) Either project 4 is selected or 5 is selected, but not both.

5) Project 3 must be selected if project 1 and 2 are selected.

6) Project 6 must be selected if project 4 and 5 are not selected.

Project Investment

1) Exactly 3 of the 6 projects should be selected.

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 3$$

2) If project 2 is selected, then project 1 must also be selected.

$$x_2 \leq x_1$$

3) Project 1 and project 3 cannot both be selected.

$$x_1 + x_3 \leq 1$$

4) Either project 4 is selected or 5 is selected, but not both.

$$x_4 + x_5 = 1$$

5) Project 3 must be selected if project 1 and 2 are selected.

6) Project 6 must be selected if project 4 and 5 are not selected.

Project Investment

1) Exactly 3 of the 6 projects should be selected.

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \ = \ 3$$

2) If project 2 is selected, then project 1 must also be selected.

$$x_2 \leq x_1$$

3) Project 1 and project 3 cannot both be selected.

$$x_1 + x_3 \ \leq 1$$

4) Either project 4 is selected or 5 is selected, but not both.

$$x_4 + x_5 = 1$$

5) Project 3 must be selected if project 1 and 2 are selected.

$$x_1 + x_2 \leq 1 + x_3$$

6) Project 6 must be selected if project 4 and 5 are not selected.

Project Investment

1) Exactly 3 of the 6 projects should be selected.

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 3$$

2) If project 2 is selected, then project 1 must also be selected.

$$x_2 \leq x_1$$

3) Project 1 and project 3 cannot both be selected.

$$x_1 + x_3 \leq 1$$

4) Either project 4 is selected or 5 is selected, but not both.

$$x_4 + x_5 = 1$$

5) Project 3 must be selected if project 1 and 2 are selected.

$$x_1 + x_2 \leq 1 + x_3$$

6) Project 6 must be selected if project 4 and 5 are not selected.

$$1 \leq x_6 + x_4 + x_5$$

Project Investment

Maximize $\boldsymbol{Z} = 16x_1 + 22x_2 + 12x_3 + 8x_4 + 11x_5 + 19x_6$

Subject to:

$5\mathrm{x}_1 + 7\mathrm{x}_2 + 4\mathrm{x}_3 + 3\mathrm{x}_4 + 4\mathrm{x}_5 + 6\mathrm{x}_6 \leq 14$ (Cash Constraint)

$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 3$ (Logical constraint #1)

$x_2 \leq x_1$ (Logical constraint #2)

$x_1 + x_3 \leq 1$ (Logical constraint #3)

$x_4 + x_5 = 1$ (Logical constraint #4)

$x_1 + x_2 \leq 1 + x_3$ (Logical constraint #5)

$1 \leq x_6 + x_4 + x_5$ (Logical constraint #6)

$x_1, \ldots, x_6 \in \{0,1\}$ (Binary constraints)

Project Investment: Python Solution

- Remember to add parameters that ensure the decision variables are binary.
- If you are having trouble figuring out how to express logical conditions constraints, use a truth table (see posted document).

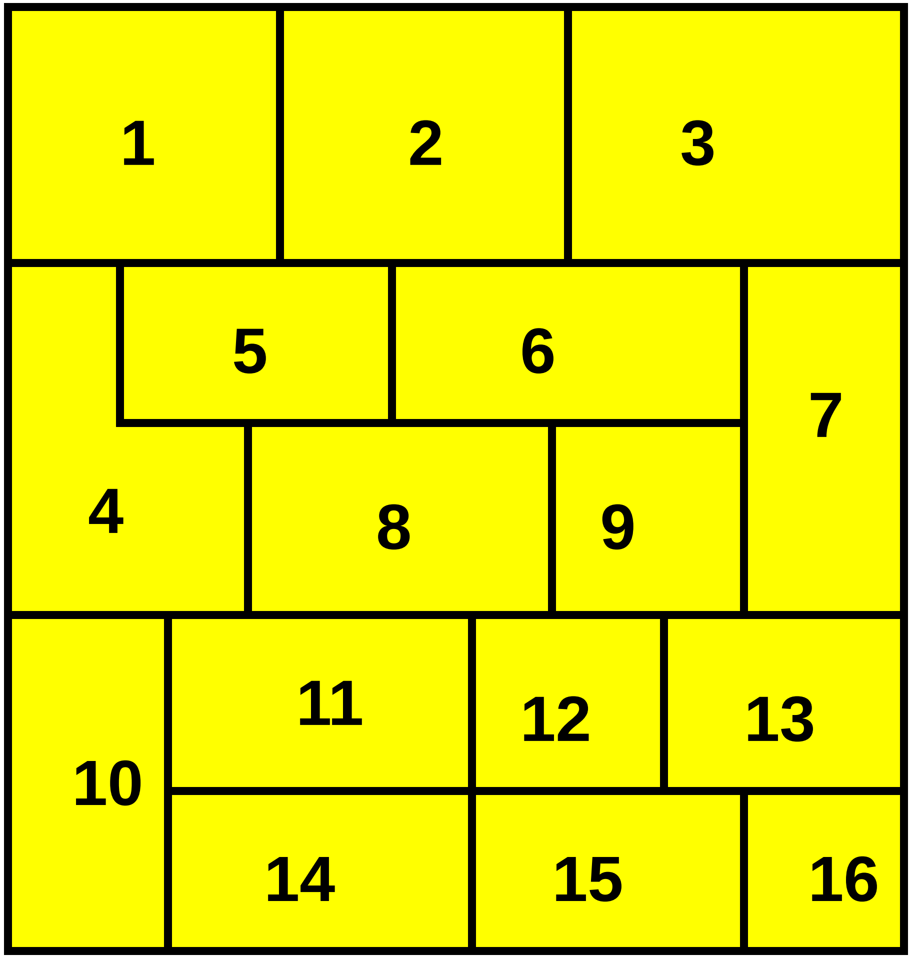
What managerial intuition do you get from the Python solution?

Covering Problems

Covering Problems

Many companies will strategically determine where to build facilities (e.g., warehouses, stores) so as to ensure that customers in certain geographic regions will frequent a particular location. For example, Starbucks® is known to open multiple stores at an intersection in order to "get to the customers on the other side." This is because "research showed that customers would travel only a few minutes to buy coffee -- or maybe six to eight minutes, tops…Even a slight bend in the road "can really have a demonstrable impact on your business in the short run." To determine where to locate these facilities, companies make assumptions regarding how much demand each facility can *cover*; they solve a **set covering** problem.

Covering Problems



Locate facilities so that each district has a facility in it or adjacent to it.

We want to build a small number of facilities as each building is costly!

Covering Problems

Define the objective

Minimize the number of facilities built

Define the decision variables

Covering Problems

Define the objective

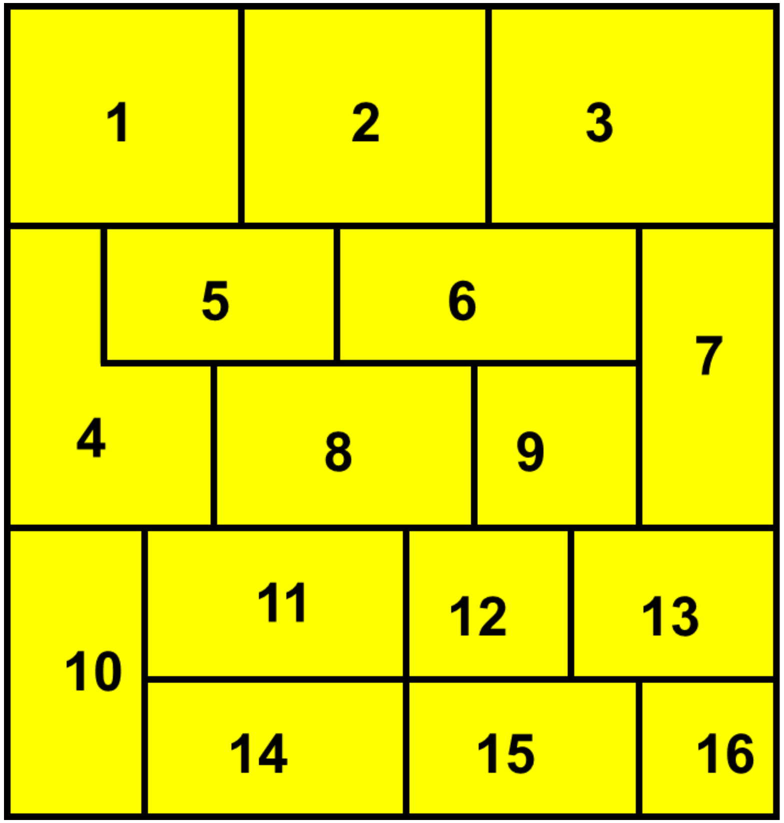
Minimize the number of facilities built

Define the decision variables

$$\boldsymbol{x_i = \begin{cases} 1, & build\ facility\ in\ location\ i\ for\ i = 1, \ldots, 16 \\ 0, & otherwise \end{cases}}$$

Covering Problems

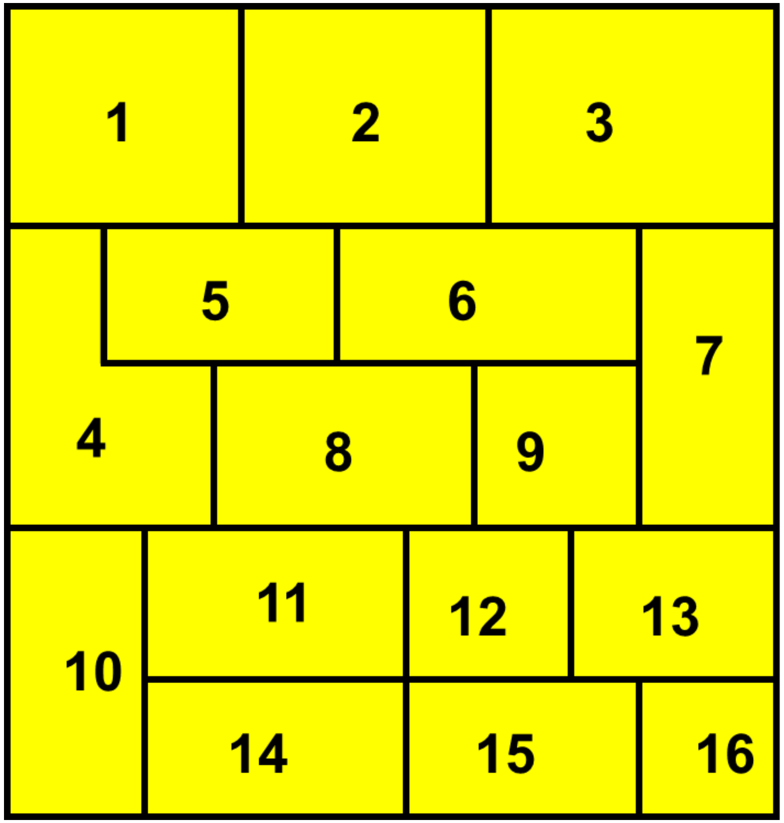
Write the mathematical objective function



Minimize $Z =$

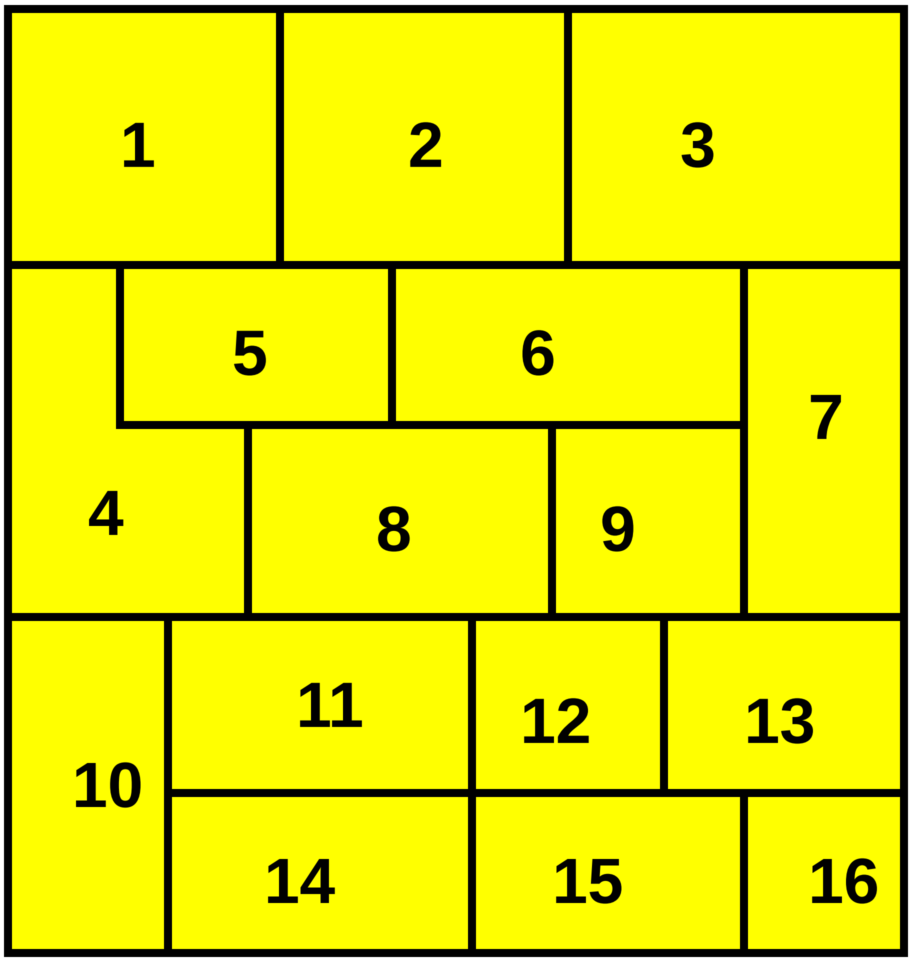
Covering Problems

Write the mathematical objective function



$$\textbf{Minimize } Z = x_1 + x_2 + \cdots + x_{16}$$

Covering Problems



District	Covers
1	1, 2, 4, 5
2	1, 2, 3, 5, 6
3	2, 3, 6, 7
⋮	⋮
16	13, 15, 16

Covering Problems

Formulating the constraints

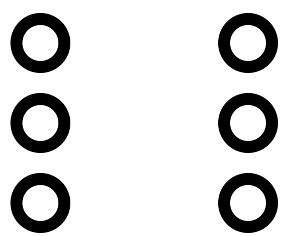
All demand must be ***covered***. Each district must have a facility in it or adjacent to it.

Covering Problems

Formulating the constraints

All demand must be ***covered***. Each district must have a facility in it or adjacent to it.

$$x_1 + x_2 + x_4 + x_5 \geq 1$$



$$x_1 + x_2 + x_3 + x_5 + x_6 \geq 1$$

$$x_{13} + x_{15} + x_{16} \geq 1$$

Covering Problems

Minimize $\boldsymbol{Z} = x_1 + x_2 + \cdots + x_{16}$

Subject to:

$x_1 + x_2 + x_4 + x_5 \geq 1$ (Covering constraint #1)

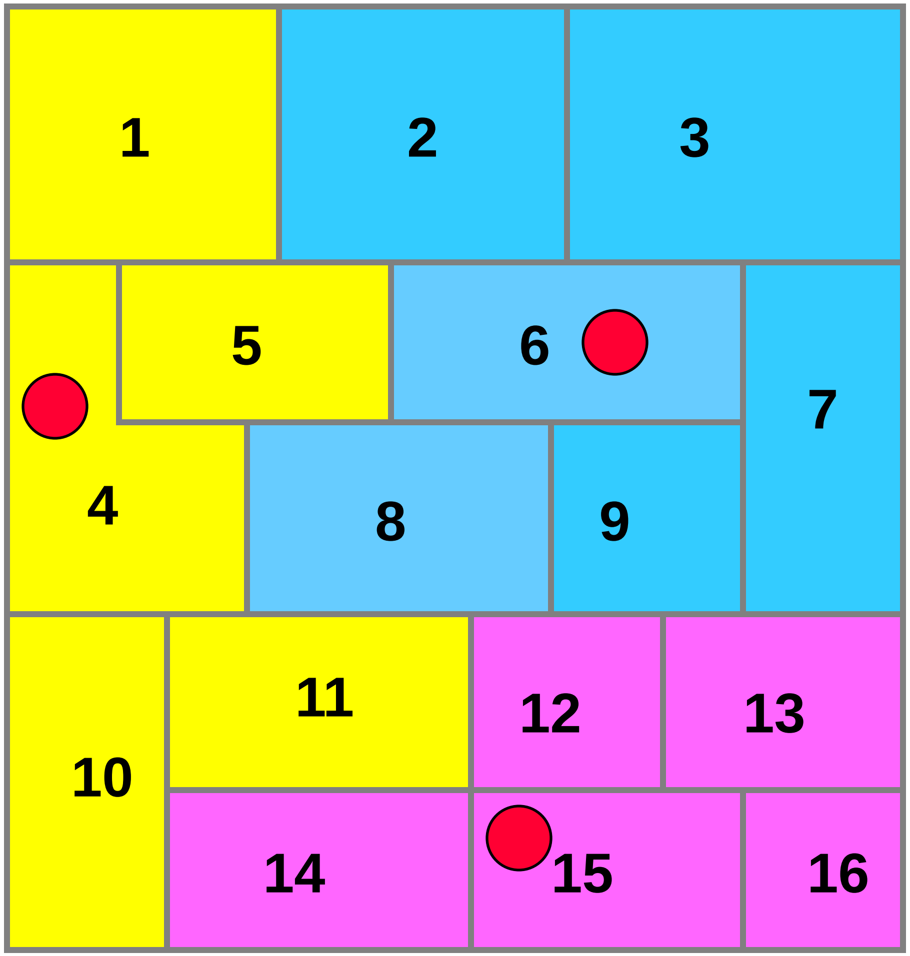
$\vdots \quad \vdots$

$x_1 + x_2 + x_3 + x_5 + x_6 \geq 1$ (Covering constraint #15)

$x_{13} + x_{15} + x_{16} \geq 1$ (Covering constraint #16)

$x_1, \ldots, x_{16} \in \{0,1\}$ (Binary constraints)

Covering Problems



District = 4

District = 6

District = 15

Total # of Facilities = 3

Covering Problems: Python Solution

- This is an example of a famous class of problems called Covering Problems which include several important special cases including the Set Covering Problem and Vertex Cover Problem. These problems are NP-hard and cannot be approximated in polynomial time to within a factor of approximately $\log n$.

What managerial intuition do you get from the Python solution?

Personnel Selection

A small architecture firm has just received confirmation that they can start designing a new building in Shanty Bay, Ontario. The firm has 8 employees and you, as a manager, have evaluated the ability of each employee to contribute to this project.

Personnel Selection

The following table summarizes the rankings of each employee to contribute to this project: 10 is the highest and 1 is the lowest.

Employee Number	Ability
1	5
2	9
3	4
4	3
5	8
6	7
7	2
8	6

The goal is to select employees so as to maximize the ability of the final group. However, there are a number of constraints.

Personnel Selection

Define the objective

Maximize the groups ability

Define the decision variables

Personnel Selection

Define the objective

Maximize the groups ability

Define the decision variables

$$x_i = \begin{cases} 1, & employee\ i\ is\ selected\ for\ i = 1, \dots, 8 \\ 0, & otherwise \end{cases}$$

Personnel Selection

Write the mathematical objective function

Employee Number	Ability
1	5
2	9
3	4
4	3
5	8
6	7
7	2
8	6

Maximize $Z =$

Personnel Selection

Write the mathematical objective function

Employee Number	Ability
1	5
2	9
3	4
4	3
5	8
6	7
7	2
8	6

Maximize $Z = 5x_1 + 9x_2 + 4x_3 + 3x_4 + 8x_5 + 7x_6 + 2x_7 + 6x_8$

Personnel Selection

Formulating the constraints

Employees 1 and 2 cannot both be selected.

Employees 4 and 5 cannot both be selected.

Personnel Selection

Formulating the constraints

Employees 1 and 2 cannot both be selected.

Employees 4 and 5 cannot both be selected.

$$x_1 + x_2 \le 1$$

$$x_4 + x_5 \le 1$$

Personnel Selection

Formulating the constraints

If employee 3 is selected, then employee 2 must be selected as well.

Personnel Selection

Formulating the constraints

If employee 3 is selected, then employee 2 must be selected as well.

$$\boldsymbol{x_3 \leq x_2}$$

Personnel Selection

Formulating the constraints

If employee 7 is selected, then employee 2 cannot be selected.

Personnel Selection

Formulating the constraints

If employee 7 is selected, then employee 2 cannot be selected.

$$x_7 \leq 1 - x_2$$

Personnel Selection

Formulating the constraints

If employee 4 is selected, then at least one of employees 5 or 6 must be selected.

Personnel Selection

Formulating the constraints

If employee 4 is selected, then at least one of employees 5 or 6 must be selected.

$$x_4 \leq x_5 + x_6$$

Personnel Selection

Formulating the constraints

If employee 6 is selected, then employees 7 and 8 must both be selected.

Personnel Selection

Formulating the constraints

If employee 6 is selected, then employees 7 and 8 must both be selected.

$$2x_6 \leq x_7 + x_8$$

Personnel Selection

Formulating the constraints

At most 3 of the employees can have an ability greater than or equal to 5.

Employee Number	Ability
1	5
2	9
3	4
4	3
5	8
6	7
7	2
8	6

Personnel Selection

Formulating the constraints

At most 3 of the employees can have an ability greater than or equal to 5.

$$\boldsymbol{x_1 + x_2 + x_5 + x_6 + x_8 \leq 3}$$

Personnel Selection

Formulating the constraints

At least 2 of the employees must have an ability less than or equal to 4.

Employee Number	Ability
1	5
2	9
3	4
4	3
5	8
6	7
7	2
8	6

Personnel Selection

Formulating the constraints

At least 2 of the employees must have an ability less than or equal to 4.

$$\boldsymbol{x_3 + x_4 + x_7 \geq 2}$$

Personnel Selection

Maximize $\boldsymbol{Z} = 5x_1 + 9x_2 + 4x_3 + 3x_4 + 8x_5 + 7x_6 + 2x_7 + 6x_8$

Subject to:

$x_1 + x_2 \leq 1$	(“Not both” employees)
$x_4 + x_5 \leq 1$	(“Not both” employees)
$x_3 \leq x_2$	(“if-one-then-both” employees)
$x_7 \leq 1 - x_2$	(“if-one-then-not” an employee)
$x_4 \leq x_5 + x_6$	(“if-one-then-another” employee)
$2x_6 \leq x_7 + x_8$	(“if-one-then-both” employees)
$x_1 + x_2 + x_5 + x_6 + x_8 \leq 3$	(Less than or equal to 3 employees)
$x_3 + x_4 + x_7 \geq 2$	(Greater than or equal to 2 employees)
$x_1, \ldots, x_8 \in \{0,1\}$	(Binary constraints)

Personnel Selection: Python Solution

- This example illustrates that logic is not only associated with formulating a constraint.
- It is important to also think about what variables should be included in a constraint.
- If the logic seems complex, another trick to simplify the problem is to add auxiliary decision variables to express intermediate steps.

What managerial intuition do you get from the Python solution?

Next Class: Mixed Integer Linear Programs

- Mixed integer-linear programming problems (MILPs) have a combination of integer, binary, and continuous decision variables in one problem.
  - There are multiple types of decision variables.
  - The constraints and the objective are linear functions of the decision variables.
  - Solutions to MILP problems are no harder to obtain than solutions to integer programming problems.
- *Big-M* constraints (i.e., the inclusion of hyper-parameters) are constraints that link one type of decision variable to another type.