Lesson 7: Logical If/Then and Either/Or Constraints

1 A Binary IP

USNA is considering purchasing some new generators to ensure power in the case of storms. The table below gives the fixed cost and MW of power generated by each potential generator.

	Fixed cost ($ million)	Power Generated (MW)
Generator 1	1.20	3,000
Generator 2	0.75	4,000
Generator 3	0.50	5,000
Generator 4	1.00	3,500
Generator 5	1.10	3,800

USNA wants to ensure that at least 8500 MW of power are available if there is an outage.

1. Formulate a concrete IP that would tell USNA the optimal generators to purchase while satisfying power demands at minimum cost.

Varrables

let $x_1 = 1$ if generator 1 is purchased and 0 otherwise

$\vdots$

let $x_5 = 1$ if generator 5 is purchased and 0 otherwise

Objective

$$\min \; 1.2x_1 + \cdots + 1.1x_5$$

Constraints

$$3000x_1 + \cdots + 3800x_5 \geq 8500$$

$$x_1, x_2, x_3, x_4, x_5 \in \{0,1\}$$

2 Types of Binary Variable Constraints

In general, there are several types of logic constraints we'd like to enforce:

- Fixed Charge (Lesson 5)
- Mutually exclusive and multiple choice constraints
- If/then constraints (binary and continuous variables)
- Either/or constraints (linear constraints on continuous variables)

3 Mutually Exclusive and Multiple Choice Constraints

The two simplest type of logical constraints deal with mutual exclusion or multiple choice.

- **Mutual Exclusive:** You are only allowed to choose a specific subset of variables to be 1.
  - Suppose in the generator problem above, you are only allowed to choose at most 1 of generators 2, 3, and 4. Write a logical constraint that enforces this.

$$\underbrace{x_2 + x_3 + x_4}_{} \leq 1$$

Only 1 of x_2, x_3 and x_4 can be 1

- **Multiple Choice:** You must choose a certain subset of variables to be equal to 1.
  - Suppose that USNA knows that facilities will only approve the purchase of exactly 2 of generators 1, 2, 3, 4, and 5. Write a logical constraint that enforces this.

$$\underbrace{x_1 + x_2 + x_3 + x_4 + x_5}_{} = 2$$

Pick exactly 2 of these variables

4 If/Then Constraints on Binary Variables

Often, it is the case that we want to enforce if/then logic on binary variables. For example, perhaps Generator 1 and 3 are made by competing companies, so if Generator 1 is purchased, we can not purchase Generator 3.

2. Try to write a constraint to capture this logic. That is write an inequality which says if $x_1 = 1$ then $x_3 = 0$.

If $x_1=1$ then $x_3=0$

Enforce this using an inequality.

$$\boxed{x_1 + x_3 \leq 1}$$

If $x_1=1$: $1 + x_3 \leq 1$

$x_3 \leq 0 \Rightarrow x_3 = 0$

If $x_1=0$

$0 + x_3 \leq 1 \Rightarrow x_3=0$ OR $x_3=1$

$x_1 + x_3 = 1$

If $x_1=1$

$1+x_3=1 \Rightarrow x_3=0$

If $x_1=0$

$0+x_3=1$

$x_3=1$

X

There's a much simpler process than guess and check that can make this work:

- Write the constraint as a conditional statement and convert all clauses to 1s

$x_3=0$ is same as $1-x_3=1$ $\Rightarrow$ If $x_1=1$ then $1-x_3=1$

- Write the constraint from left to right.

- Variable before then is LHS constraint
- Then is $\leq$
- Variable after then is RHS constraint

$\Rightarrow x_1 \leq 1-x_3$

- **ALWAYS** check every logical constraint for **explicit** and **implicit** satisfaction.
  - **Explicit** satisfaction:

- Does it satisfy the if/then?

$x_1=1 \Rightarrow 1 \leq 1-x_3$

$x_3=0$ ✓

  - **Implicit** satisfaction:

- If LHS not true is RHS not restricted?

$x_1=0 \Rightarrow 0 \leq 1-x_3$

x_3 free ✓

Using the steps above, capture the following if/then constraint logic:

3. If we purchase generator 2 then we must purchase generator 4.

(1) If $x_2=1$ then $x_4=1$

(2) Convert all to $=1$ Done

(3) $x_2 \le x_4$

(4) $x_2=1$

$1 \le x_4$

$x_4=1$ ✓

$x_2=0$

$0 \le x_4$

$x_4=0$ OR $x_4=1$

x_4 free ✓

4. If we don't purchase generator 1 then we must purchase generator 5.

(1) If $x_1=0$ then $x_5=1$

(2) If $(1-x_1)=1$ then $x_5=1$

(3) $1-x_1 \le x_5$

(4) $x_1=0$

$1-0 \le x_5$

$x_5=1$

$x_1=1$

$1-1 \le x_5$

$x_5 \ge 0$

x_5 free

3 or more variables this gives a starting point

The same idea applies with 3 or more variables, but it does get more complicated and you have to think through the logic. We'll try two more:

5. If we purchase generator 2 or generator 1 then we can't purchase generator 5.

① If $x_2=1$ or $x_1=1$ then $x_5=0$

② If $x_2=1$ or $x_1=1$ then $(1-x_5)=1$

$x_2=1 \rightarrow x_5=0$
$x_1=1 \rightarrow x_5=0$] Together Satisfy the OR

$x_1=1$ and $x_2=1 \rightarrow x_5=0$

③ $x_2 \le (1-x_5)$
$x_1 \le (1-x_5)$

④ $x_2=1 \rightarrow 1 \le (1-x_5)$
$x_5=0$
$x_1=1 \rightarrow 1 \le (1-x_5)$
$x_5=0$ ✓

$x_1=0 \quad x_2=0$
$0 \le (1-x_5)$
$0 \le (1-x_5)$
x_5 free ✓

6. If we don't purchase generator 1 and we do purchase generator 3 then we must also purchase generator 2.

① If $x_1=0$ and $x_3=1$ then $x_2=1$

② If $(1-x_1)=1$ and $x_3=1$ then $x_2=1$

③ $1-x_1+x_3 \le x_2$

④ $x_1=0, x_3=1$
$1-0+1 \le x_2$
$x_2 \ge 2$ doesn't work

Try replacing x_2 with $2 \cdot x_2$
$1-x_1+x_3 \le 2x_2$

Check: $1-0+1 \le 2 \cdot x_2 \rightarrow x_2=1$

Try replacing x_2 with x_2+1

$$\boxed{1-x_1+x_3 \le x_2+1}$$

Check:
$1-0+1 \le 1+x_2 \rightarrow x_2=1$

Implicity:
$x_1=1, x_3=1 \quad 1-1+1 \le x_2+1 \rightarrow x_2$ free
$x_1=1, x_3=0 \quad 1-1+0 \le x_2+1 \rightarrow x_2$ free
$x_1=0, x_3=0 \quad 1-0+0 \le x_2+1 \rightarrow x_2$ free

Implicit: $x_1=1$ $x_3=1$

$1-1+1 \leq 2x_2$ Doesn't work

$2x_2 \geq 1 \to x_2=1$

5 Either/Or Constraints: Motivating Example

Now, we switch gears to another application of binary variables.

Quality Cabinets used an integer-program to determine how many of each type of cabinet to make in order to maximize profit. They used decision variables x_s, x_d, and x_e to represent the number of standard, deluxe, and enhanced cabinets, respectively, to produce each week. Each cabinet requires a certain number of hours of painting time and there is a limit on the number of painting hours available. A small part of the model is:

Maximize	$25x_a + 45x_d + 60x_e$	
subject to	$2x_a + 4x_d + 5x_e \leq 700$	(painting time)

6 Model Update Requested

Now Quality Cabinets is considering renting better painting equipment and has asked us to update our model to help with this decision. The equipment will cost \$300 per week, but will reduce the time required to to do the painting by 15 minutes for a standard cabinet, by 30 minutes for a deluxe cabinet, and by 1 hour for an enhanced cabinet.

We decide to add a binary variable z. In the solution, if $z = 1$, then Quality Cabinets should rent the equipment. If $z = 0$, they should not.

7. Should the objective function change? If not, explain. If so, write the updated objective function.

$Z=0$ don't pay \$300

$Z=1$ pay \$300

Max $25x_a + 45x_d + 60x_e - 300 \cdot Z$

8. If the equipment is not rented, what should the painting constraint be? Label it (A).

(A) $2x_a + 4x_d + 5x_e \leq 700$

If $Z=0$ don't buy machine keep original constraint

9. If the equipment is rented, what should the painting constraint be? Label it (B).

(B) $1.75x_a + 3.5x_d + 4x_e \leq 700$

If $Z=1$ buy machine

Don't do this: $(2-0.25z)x_a + (4-0.5z)x_d + (5-1z)x_e \leq 700$

Nonlinear doesn't work

7 Logical (Either/Or) Constraints

There is no place for "if – then" statements in a math programming model, so we have to enforce this logic indirectly using linear constraints and binary variables.

> A constraint is *relaxed* if it has no impact. We can relax a "$\leq$" constraint by making the value on the right so large that it will never restrict the values of the variables on the left.

10. Rewrite constraint (A) above *using* z so that it is enforced if $z = 0$ (the equipment is not rented), but *relaxed* if $z = 1$ (the equipment is rented).

(A) $2x_a + 4x_d + 5x_e \leq 700 + Mz$

$z=0$: Don't buy RHS $700 + M \cdot 0$ enforced
$z=1$: Buy RHS $700 + M \cdot 1$ Relaxed

11. Rewrite constraint (B) above *using* z so that it is enforced if $z = 1$ (the equipment is rented), but *relaxed* if $z = 0$ (the equipment is not rented).

(B) $1.75x_a + 3.5x_d + 4x_e \leq 700 + M(1-z)$

$z=0$ Don't buy RHS $700 + M \cdot 1$
$z=1$ Buy RHS $700 + M \cdot 0$

12. Write the updated version of the partial Quality Cabinets model here.

Max $25x_a + 45x_d + 60x_e - 300 \cdot z$

(A) $2x_a + 4x_d + 5x_e \leq 700 + Mz$

(B) $1.75x_a + 3.5x_d + 4x_e \leq 700 + M(1-z)$

$z=0$: Don't buy Pay 0, A enforced, B relaxed
$z=1$: Buy Pay 300, A relaxed, B enforced

$a^T x \le b$

8 Either/Or Constraints Summary

Suppose $a^T x \le b$ is a linear constraint, M is a number that is bigger than $a^T x$ could ever be (but not TOO big!–choose M wisely), and z is a binary variable.

A constraint that is enforces $a^T x \le b$ if $z = 0$ and relaxes $a^T x \le b$ if $z = 1$:

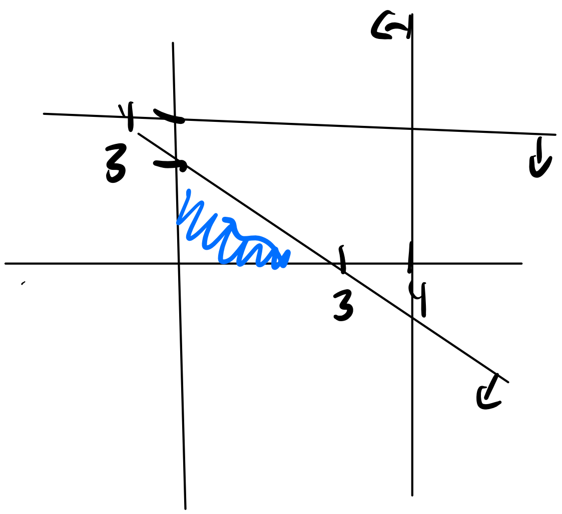
$$a^T x \le b + M \cdot z$$

A constraint that is enforces $a^T x \le b$ if $z = 1$ and relaxes $a^T x \le b$ if $z = 0$:

$$a^T x \le b + M(1-z)$$

9 Many more possibilities

There are many ways to creatively use linear constraints to enforce modeling requirements; this lesson contains only a few examples. Often this process takes some trial and error. **Always be sure to test the logic with various values of the decision variables to make sure the constraint is doing what you want it to do.**



$x \le 4$

$y \le 4$

$x + y \le 3$

If I push constraint $x+y \le 3$ up and to the right it won't affect values of x and y.

Add number to RHS.

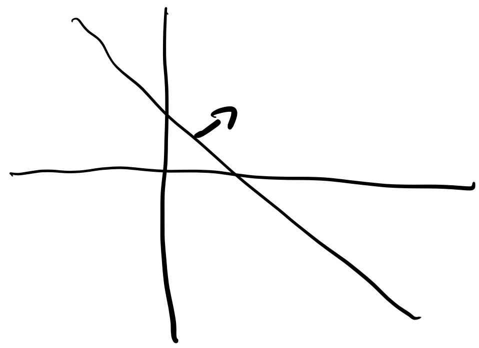
M

$x + y \le 3 + M$

Relaxing $x+y \le 3$

$$a^T x \geq b$$

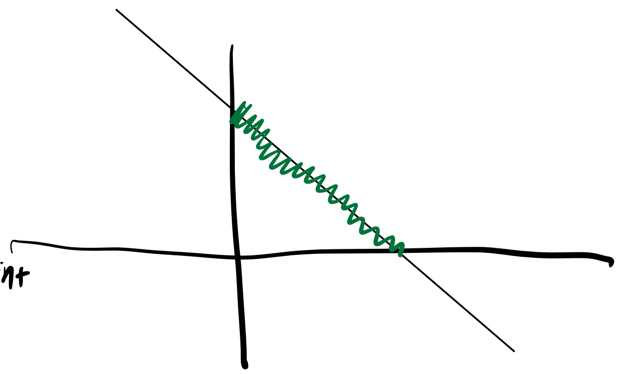
Relaxed if $a^T x \geq b - M$



Enforce if $z=0$

$$a^T x \geq b - M \cdot z$$

$$a^T x = b$$



Relaxed equality constraint

$$a^T x \leq b + M$$
$$a^T x \geq b - M$$

Relaxed if $z=0$ $\quad a^T x = b$

$$a^T x \leq b + M(1-z)$$
$$a^T x \geq b - M(1-z)$$