

Lesson 8: 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.

Variables \rightarrow whether USNA buys a generator or not

let $x_i = 1$ if USNA purchases generator i or 0 otherwise for $i \in \{1, 2, 3, 4, 5\}$

Objective

$$\min \text{cost: } 1.2x_1 + 0.75x_2 + 0.5x_3 + 1x_4 + 1.1x_5$$

Constraints

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

$$3000x_1 + 4000x_2 + 5000x_3 + 3500x_4 + 3800x_5 \geq 8500$$

2 Types of Binary Variable Constraints

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

- Fixed Charge (Lesson ⁶●) → Strong and weak forcing
 - Mutually exclusive and multiple choice constraints
 - If/then constraints
 - Either/or constraints
- Hard / tricky constraints

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.

Can choose at most 1 of gen 2, 3, 4

$$x_2 + x_3 + x_4 \leq 1$$

How many of generators 2, 3, and 4 are chosen

At most 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.

Must Choose exactly 2 of gen 1, 2, 3, 4, 5

$$x_1 + x_2 + x_3 + x_4 + x_5 = 2$$

How many of the generators do I choose

Exactly 2

4 Introducing If/Then Constraints on Two 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. x_1, x_3 binary If $x_1=1$ then $x_3=0$

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

$$x_1 + x_3 = 1$$

$$x_1=1 \rightarrow x_3=0 \quad \checkmark$$

$$x_1=0 \rightarrow x_3=1 \quad \times$$

Doesn't work because if $x_1=0$ then x_3 is restricted

$$x_1 \cdot x_3 = 0$$

$$x_1=1 \rightarrow x_3=0$$

$$x_1=0 \rightarrow x_3 \text{ free}$$

Works Mathematically but is nonlinear

$$x_1 + x_3 \leq 1$$

$$x_1=1 \rightarrow x_3=0 \quad \checkmark$$

$$x_1=0 \rightarrow x_3 \leq 1$$

x_3 free

Mathematically works and is linear ☺

$$x_1 \leq 1 - x_3$$

$$x_1=1 \rightarrow x_3=0$$

$$x_1=0 \rightarrow 1 - x_3 \geq 0$$

x_3 free

Works

It can seem like a lot of guess and check in order to get a constraint that actually works. Fortunately, there's a process you can follow to make this significantly easier. Namely, the following steps can make writing these logical constraints much easier.

For if/then on binary variables If $x_1=1$ then $x_3=0$

- Write the constraint as a conditional statement and convert all clauses to 1s (why?)

$$\text{If } x_3=0 \rightarrow (1-x_3)=1$$

$$\underbrace{\text{If } x_1=1}_{\text{LHS}} \underbrace{\text{then}}_{\leq} \underbrace{(1-x_3)=1}_{\text{RHS}}$$

- Write the constraint from left to right.

- The left side of the **if** statement is the LHS of the constraint

- The inequality is always \leq] then $\rightarrow \leq$

- If it is converted to 1s, everything following **then** is the RHS of the constraint. That is the high value on the left "pushes" the RHS to its high value.

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

- Explicit** satisfaction: Does the conditional statement work?

$$x_1=1 \rightarrow 1 \leq 1-x_3 \rightarrow x_3=0 \quad \checkmark$$

- Implicit** satisfaction: If LHS is not true I can't restrict the value of the RHS.

$$x_1=0 \rightarrow 0 \leq 1-x_3 \rightarrow x_3=1 \quad \checkmark \quad x_3 \text{ is free}$$

3

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

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

① $\underbrace{\text{If } x_2=1}_{\text{LHS}} \underbrace{\text{then}}_{\leq} \underbrace{x_4=1}_{\text{RHS}}$

③ Explicit check

$x_2=1 \rightarrow 1 \leq x_4 \rightarrow x_4=1 \checkmark$

② $x_2 \leq x_4$

④ Explicit check

$x_2=0 \rightarrow 0 \leq x_4 \rightarrow x_4 \text{ free}$

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

① $\text{If } \underline{x_1=0} \text{ then } x_5=1$

$\text{If } \underline{(1-x_1)=1} \text{ then } \underline{x_5=1}$

③ $x_1=0 \rightarrow 1 \leq x_5 \rightarrow x_5=1 \checkmark$

④ $x_1=1 \rightarrow 1-1 \leq x_5$
 $x_5 \geq 0 \rightarrow \text{free} \checkmark$

② $1-x_1 \leq x_5$

5. If we don't purchase generator 3 then we can't purchase generator 1. and we must buy gen 4.

① $\text{If } x_3=0 \text{ then } x_1=0 \text{ and } x_4=1$

$\underbrace{\text{If } (1-x_3)=1}_{\text{LHS}} \underbrace{\text{then}}_{\leq} \underbrace{(1-x_1)=1 \text{ and } x_4=1}_{\text{RHS}}$

③ $x_3=0$

$1 \leq (1-x_1) + x_4 \quad \times$

$2 \leq (1-x_1) + x_4$
 $x_1=0, x_4=1 \checkmark$

② $1-x_3 \leq (1-x_1) + x_4$
 $2(1-x_3) \leq (1-x_1) + x_4$

④ $x_3=1$

$0 \leq (1-x_1) + x_4$
 $x_1 \text{ and } x_4 \text{ are free} \checkmark$

The same idea will also work with 3 or more variables, just the LHS or RHS of the constraints may be off by a constant factor. That's why it's important to always check explicit and implicit satisfaction.

5 Either/Or Constraints: Motivating Example

The remainder of this lesson will focus on either/or constraints.

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

x_s : standard
 x_e : enhanced
 x_d : deluxe

$$\underline{2}x_s + \underline{4}x_d + \underline{5}x_e \leq 700$$

Time to paint each cabinet

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 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.

and reduce the painting time

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

$$\text{Max } \underbrace{25x_s + 45x_d + 60x_e}_{\text{Revenue}} - \underbrace{300z}_{\text{Fixed cost}}$$

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

If machine is not bought

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

$z=0$

Constraint

stays same

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

If equipment
rented $z=1$
 x_a reduced 15 min
 x_d reduced 30 min
 x_e reduced 1 hr

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

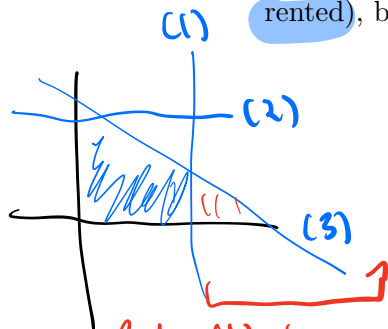
If $z=0$ constraint (A)
 $z=1$ constraint (B)

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.

9. 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).



Relax (1) by
adding a number
to push out
of feasible region

$$(A) \quad 2x_a + 4x_d + 5x_e \leq 700 + M \cdot z$$

$$z=0: \leq 700 \text{ enforced } \checkmark$$

$$z=1: \leq 700 + M \text{ Relaxed } \checkmark$$

\nearrow
A big
number

10. 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).

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

$$z=0: \leq 700 + M \text{ Relaxed } \checkmark$$

$$z=1: \leq 700 \text{ Enforced } \checkmark$$

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

$$\begin{aligned} \max \quad & 25x_a + 45x_d + 60x_e - 300z \\ \text{s.t.} \quad & 2x_a + 4x_d + 5x_e \leq 700 + M \cdot z \quad (A) \\ & 1.75x_a + 3.5x_d + 4x_e \leq 700 + M(1-z) \quad (B) \end{aligned}$$

$z=0$: Don't buy
(A) enforced
(B) Relaxed

$z=1$ Buy pay 300
(A) Relaxed
(B) Enforced

8 Logical Constraints Summary

M should be logical for problem \rightarrow sum of demand
sum of supply
Capacity etc

Suppose $a^T x \leq b$ 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 enforces $a^T x \leq b$ if $z = 0$ and relaxes $a^T x \leq b$ if $z = 1$:

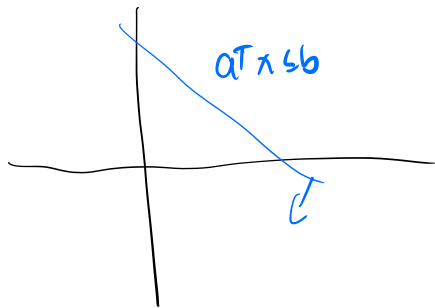
$$a^T x \leq b + M \cdot z$$

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

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

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.**



$a^T x \leq b + M$ relaxes this constraint



$a^T x \geq b - M$ relaxes the constraint

$a x \geq b \rightarrow a x \leq b$

$$(2 - 0.25z) x_5$$

$$2x_5 - 0.25 \underbrace{zx_5}$$