MODELING WITH BINARY VARIABLES

Class 3 – October 1, 2025

Context

- You have several projects available A, B, ...,
- You choose **which projects to fund**
- A=1 **if and only if** project A is funded

If you fund **A**, you should also fund **E**

- What are the feasible values for A, E?
  - Recall that A, E are **binary**
  - We want: *if A=1, must have E=1*

ALL OPTIONS:

A	**E**	
0	0	✓
0	1	✓
~~1~~	~~0~~	
1	1	✓

- How about: **A** $\leq$ **E**
  - If A=1, the only option is E=1
  - If A=0, can set any value for E

- **Remember!** "If you fund **A**, then you should fund **B**": **A** $\leq$ **B**

- **Q:** "If you do **not** fund **A**, then you should fund **B**"
  - Add a constraint: $1 - A \leq B$
    - "Not selecting A" is same as $1 - A = 1$, so this is just like **Q5 !**

Logical Implications with Binary Variables

- **Q**. If you fund project A, then you should fund projects E **and** H.
  - Same as: “If you fund A, then fund E” and “If you fund A, then fund H”
  - A <= E, A <= H
  - Also possible to do this with **one** constraint: A <= (E+H)/2

  Q. Why not A <= E+H?

- **Q**. If you fund anything from **A/B/C**, then also fund **H**.
  - Same as: “If you fund A, then fund H” and “If you fund B, then fund H”, …
  - A <= H, B <= H, C <= H
  - Also possible to do this with **one** constraint: (A+B+C)/3 <= H

  Q. Why not A + B + C <= H?

General Recipe for Defining Indicators

$$\mathbf{Y = 1} \textit{ if and only if } \mathbf{a_1 X_1 + \ldots + a_n X_n + b \geq 0}$$

- **Y** is a binary decision variable, $X_1, \ldots, X_n$ are continuous or discrete decisions
- $a_1, \ldots, a_n, b$ are parameters/data

- **The first implication:**

(1): *If* $\mathbf{Y = 1}$ *then* $\mathbf{a_1 X_1 + \ldots + a_n X_n + b \geq 0}$

- This is equivalent to the following linear constraint:

$$\mathbf{a_1 X_1 + \ldots + a_n X_n + b \geq m} \cdot (1 - \mathbf{Y})$$

- *In practice, '**m**' is the smallest value that* $a_1 X_1 + \ldots + a_n X_n + b$ *can take*

- **Understand why this works. No need to remember the constraint!**

General Recipe for Defining Indicators

$$\mathbf{Y = 1} \ \textit{if and only if} \ \mathbf{a_1 X_1 + \ldots + a_n X_n + b \geq 0}$$

- **Y** is a binary decision variable, $\mathbf{X_1, \ldots, X_n}$ are continuous or discrete decisions
- $\mathbf{a_1, \ldots, a_n, b}$ are parameters/data

- **The first implication:**

  (1): *If* $\mathbf{Y = 1}$ *then* $\mathbf{a_1 X_1 + \ldots + a_n X_n + b \geq 0}$

- In practice, you can directly implement **(1)** in Gurobi with:

  model.addGenConstrIndicator(Y, ***True****,* $a_1 X_1 + \ldots + a_n X_n + b \geq 0$ *)*

  Syntax: *model.addGenConstrIndicator(* **Y***,* **boolean value***,* **implied (in)equality***)*
  - **Y** = a Gurobi binary variable
  - **boolean value** = True or False
  - **implied (in)equality** = linear relationship that should hold when Y = **boolean value**

  This implements **one** direction: *"If Y=boolean value, then implied (in)equality"*
  https://www.gurobi.com/documentation/current/refman/py_model_agc_indicator.html

General Recipe for Defining Indicators

$$\mathbf{Y = 1} \textit{ if and only if } \mathbf{a_1 X_1 + \ldots + a_n X_n + b \geq 0}$$

- Y is a binary decision variable, $X_1, \ldots, X_n$ are continuous or discrete decisions
- $a_1, \ldots, a_n, b$ are parameters/data

- **The second implication:**

  (2) *If* $Y = 0$ *then* $a_1 X_1 + \ldots + a_n X_n + b < 0$

- Because we cannot have **strict** inequality < 0, instead we implement:

  If $Y = 0$ *then* $a_1 X_1 + \ldots + a_n X_n + b \leq -\varepsilon$

  - If $X_1, \ldots, X_n$ are integer, reformulation can be made exact. Otherwise, take '**ε**' as a small tolerance (e.g., 0.00001).

- Implemented with: $a_1 X_1 + \ldots + a_n X_n + b + \varepsilon \leq (M + \varepsilon) Y$
  - *In practice, '**M**' is the largest value that* $a_1 X_1 + \ldots + a_n X_n + b$ *can take*

Recap

$$Y = 1 \textit{ if and only if } a_1 X_1 + \ldots + a_n X_n + b \geq 0$$

Y is a binary decision variable, $X_1, \ldots, X_n$ are continuous or discrete decisions

$a_1, \ldots, a_n, b$ are parameters/data

(1): *If* $Y = 1$ *then* $a_1 X_1 + \ldots + a_n X_n + b \geq 0$

(2): *If* $Y = 0$ *then* $a_1 X_1 + \ldots + a_n X_n + b < 0$

Implemented with linear constraints:

(1) $a_1 X_1 + \ldots + a_n X_n + b \geq m \cdot (1 - Y)$

(2) $a_1 X_1 + \ldots + a_n X_n + b + \varepsilon \leq (M + \varepsilon)\, Y$ (ε=1 if $X_1, \ldots, X_n$ integer)

In Gurobi:

(1) *model.addGenConstrIndicator(* Y*, True,* $a_1 X_1 + \ldots + a_n X_n + b \geq 0$ *)*

(2) *model.addGenConstrIndicator(* Y*, False,* $a_1 X_1 + \ldots + a_n X_n + b \leq -\varepsilon$ *)*

"Cheat-Sheet"

X and **Y** are decisions; **a**, **b** are parameters/data; **a X** denotes any linear expression in **X**

1. (X,Y bin) ***"If X = 1 then Y = 1"*** → add constraint: $X \le Y$
2. (X,Y bin) ***"If X = 1 then Y = 1, and vice-versa"*** → add constraint: $X = Y$

3. (Y bin) ***"If Y = 1 then a X + b ≥ 0"*** → add constraint: $a X + b \ge m \cdot (1-Y)$
   - 'm' is the *smallest* value $a X + b$ can take
4. (Y bin) ***"If Y = 1 then a X ≥ b"*** → add constraint: $a X - b \ge m \cdot (1-Y)$
   - 'm' is the *smallest* value $(a X - b)$ can take
5. (Y bin) ***"If Y = 1 then a X ≤ b "*** → add constraint: $a X - b \le M \cdot (1-Y)$
   - 'M' is the *largest* value $(a X - b)$ can take
6. (Y bin) ***"If Y = 1 then a X + b ≤ 0 "*** → add constraint: $a X + b \le M \cdot (1-Y)$
   - 'M' is *largest* value $(a X + b)$ can take

3-6 are all "the same"! Use whichever you like!

7. (Y bin) ***"If Y = 1 then a X + b > 0 "*** → **CAN'T DO > 0.**
   - Instead, do ***"If Y = 1 then a X + b ≥ ε" for a very small number ε > 0***
   - To implement, add the constraint: $a X + b - \varepsilon \ge (m - \varepsilon)(1-Y)$, where 'm' is the smallest value $(a X + b)$ can take

8. If you need ***"If Y = 0 then …"***, replace *Y* in the constraint with $1-Y$
9. If you need ***"If a X + b ≤ 0 then Y = 1"***, replace this with ***"If Y=0, then a X + b > 0"***

10. (Y bin) Need **"X * Y"** → add new variable **Z** ("= X * Y") and constraints:

$$Z \le M \cdot Y \qquad Z \ge m \cdot Y \qquad Z \le X - m \cdot (1 - Y) \qquad Z \ge X - M \cdot (1 - Y)$$

   - m/M are smallest/largest value that X can take