Integer Programming

- Enumerative solution n^p possibilities (p → # of decision vars)
  - 2^p (for binary)
  - {0 to n} → Int.
- Logical constraints → Gates A — B $d_B \le d_A$
  - On binary variables
  - → get in subspace
    $d_i + d_j + d_k \le 2$
    If only 2 from i,j,k
    Total d → $d_a, d_b, \dots d_i, d_I$
- Big M problem
  - Integer/Continuous + Binary
    - → how many outputs.
      - # of items
    - Establish something
      - Ex. factory Y/N

Can be any #

B_i 0/1 $\qquad O_i$

$B_i \times B_i < O_i \le M \times B_i$

- M → Constant for all B_i's O_i's
- M → Determined after, assuming Upper Bound of list (O_i's)

⇒ Objective → $e_1 \times 50K + e_2 \times 55K + e_3 \times 68K +$

Integer Programming PRACTISE :-

Ferari wants to manufacture a car :-

↳ 3 engine choices → 2 pseudo choices
+ 1 turbo engine choice
+ 2 break choices
+ 2 interior choices.

Decision vars. :- $e_1, e_2, e_3, te_1, b_1, b_2, i_1, i_2$

↳ All are binary decisions. → $\in [0, 1]$

~~Constraints~~ →

1. Cost associated objective

⇒ $e_1 \times 50K + e_2 \times 55K + e_3 \times 65K + te_1 \times 5K +$
$b_1 \times 5K + b_2 \times 7K + i_1 \times 15K + i_2 \times 20K \leq = obj$

↳ Minimize.

2. Constraints

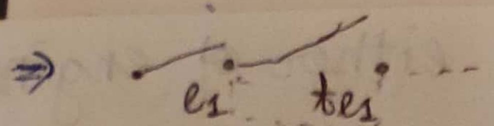
1. Horsepower constraint :- (1)

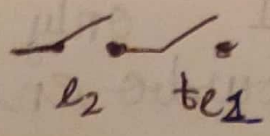
$e_1 \times 430 + e_2 \times 600 + e_3 \times 750 + te_1 \times 200 \geq 600$

2. Association of engines.

↳ e1 and e2 with te1

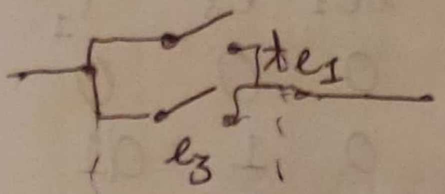
↳ e3 and te2 never together.

⇒ 



$te_1 \leq e_1$ — ②

$te_1 \leq e_2$ — ③



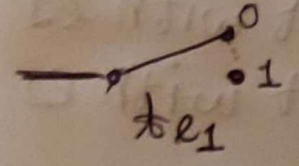
$te_1 + e_3 \leq 1$ — ④

↳ Alway another line ∴ both te_1 and $e_3 == 0$

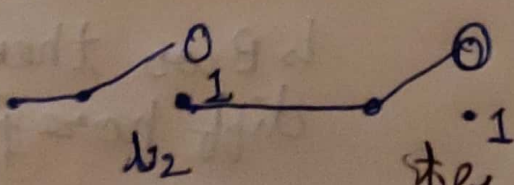
3. Only 1 primary engine allowed ÷
↳ $e_1 + e_2 + e_3 = 1$ — ⑤

4. Only 1 break allowed ÷
↳ $b_1 + b_2 = 1$ — ⑥

5. Break and engine combination ÷ पिछे हुए.
If turbo engine then ~~break~~ b_2.
te_1 (ये हो चुका होना चाहिए).



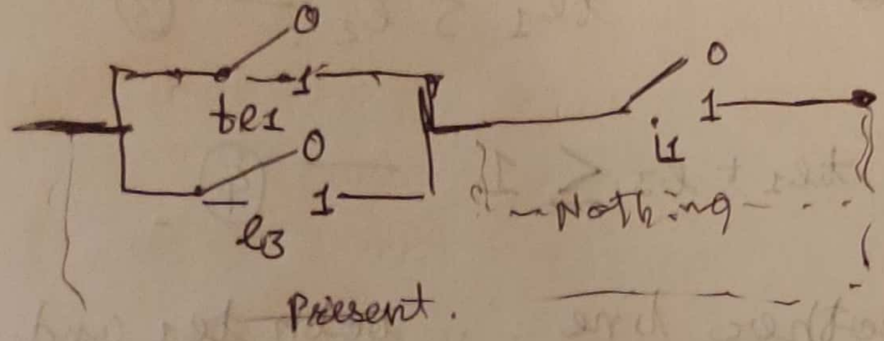
for this we require $b_2 \rightarrow$ ON.



$te_1 \leq b_2$ — ⑦

6. Exactly 1 interior.
$i_1 + i_2 = 1$ — ⑧

7. Interior 1 only when either of engine 3 or turbo engine.



te_1	e_3	i_1
0	0	0
0	1	0/1
1	0	0/1
1	1	0/1

$te_1 + e_3$	i_1
0	0
1	0/1
1	0/1
2	0/1

and = Multiplication
or = Addition – ✓

$i_1 \leq te_1 + e_3$ — (9)

Mistakes :-

1. There are two dec. vars for turbo eng.
   - ↳ Even if we have got 1 entity, we can have two d. vars
     - ↳ To use it with e1
     - \+ To use it with e2
       - ↳ Bcos they use diff. horse power.

Pseudo 2 turbo engines. ← te_1, te_2

Obj → A[[0,1] [2,5]]
↳ A[0,2]   A[1,5]

np.ix_([1,2,3], [2,5])
↳ [12, 15
22, 25
32, 35]

A[1, [2,5]] → Broadcasted
↳ A[1,2], A[1,5]

But A[[1,2,3] [2,5]]
X – Fails.

⇒ Objective → $e_1 \times 50K + e_2 \times 55K + e_3 \times 68K + te_1 \times 8K + te_2 8K + b_1 \times 5K, b_2 \times 7K + i_1 \times 18K + i_2 \times 20K$.

C·1 → $e_1 \times 480 + e_2 \times 600 + e_3 \times 750 + te_1 \times 200 + te_2 \times 150 \geq 600$

0 1 2 3 4 5 6 7 8

↳ For a dict if required.

C·2 → $te_1 \leq e_1$.

C3 → $te_2 \leq e_2$.

C4 → $te_1 + te_2 + e_3 \cdot \leq 1$.

C5 → $e_1 + e_2 + e_3 = 1$.

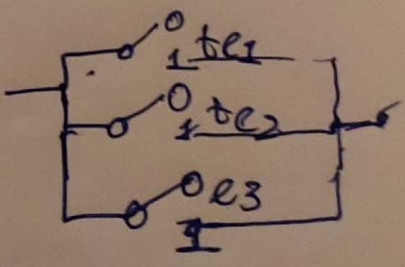
C6 → $b_1 + b_2 = 1$.

C7 → $te_1 \leq b_2 + te_2 \leq b_2$

↳ $te_1 + te_2 \leq 2b_2$

C8 → $i_1 + i_2 = 1$

C9 → $i_1 \leq te_1 + te_2 + e_3$.



D·vars → e_1 e_2 e_3

te_1 te_2

b_1 b_2

i_1 i_2.