REOP - Class 6 (Integer Programming)

I/ Homework & reminders

Shortest paths :

Settings	Algorithms
Directed Acyclic Graph	Topological ordering + DP
No negative edges	Dijkstra
No negative cycles	Bellman - Ford

Flows & Bellman → other class ?

Complexity : proving NP - completeness / hardness

P1 : Hamiltonian Circuit
Given $G_1 = (V_1, E_1)$, is there a circuit visiting each vertex exactly once ?

decision problem

P2 : Traveling Salesman Problem
Find a traveling salesman tour of minimum cost in a complete weighted graph $G_2 = (V_2, E_2)$ w_2

optim problem

Prove that P2 is NP-hard, knowing that D1 is NP-complete

D2 : TSP - Decision version
Is there a tour of cost $\leq W_2$? (for fixed W_2)

decision problem

$\underline{P2} \geq D2$

Catalogue of decision pbs with complexity

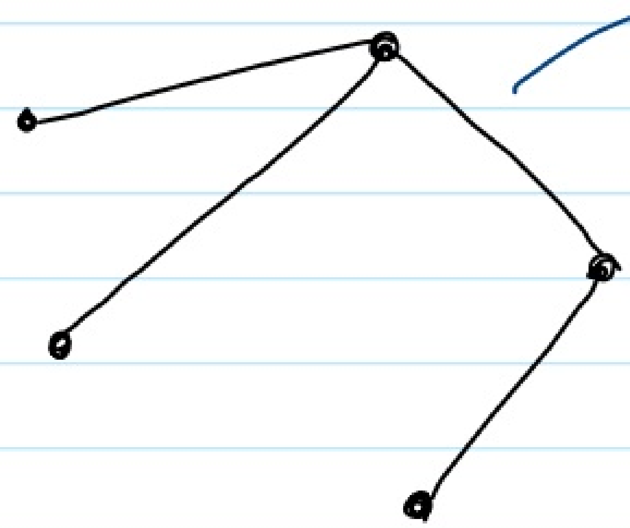
We need to find a polynomial reduction from D1 to D2 to prove $D2 \geq D1$

Given an instance to Hamiltonian Circuit (G_1)
build an instance to TSP-Decision (G_2)

We define weights $w_2(e) = \begin{cases} 1 & \text{if } e \in E_1 \\ 2 & \text{if } e \notin E_1 \end{cases}$

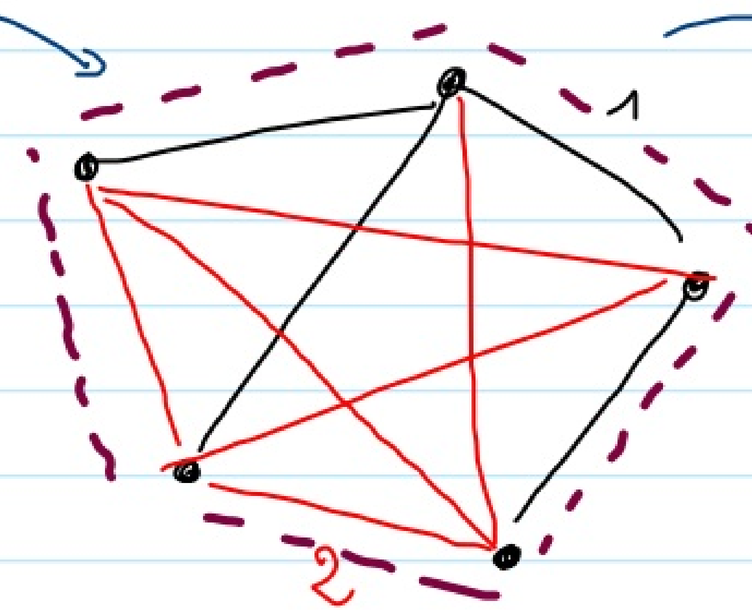
This reduction φ is polynomial (enumerate over V_1 & E)

G_1



reduction

G_2

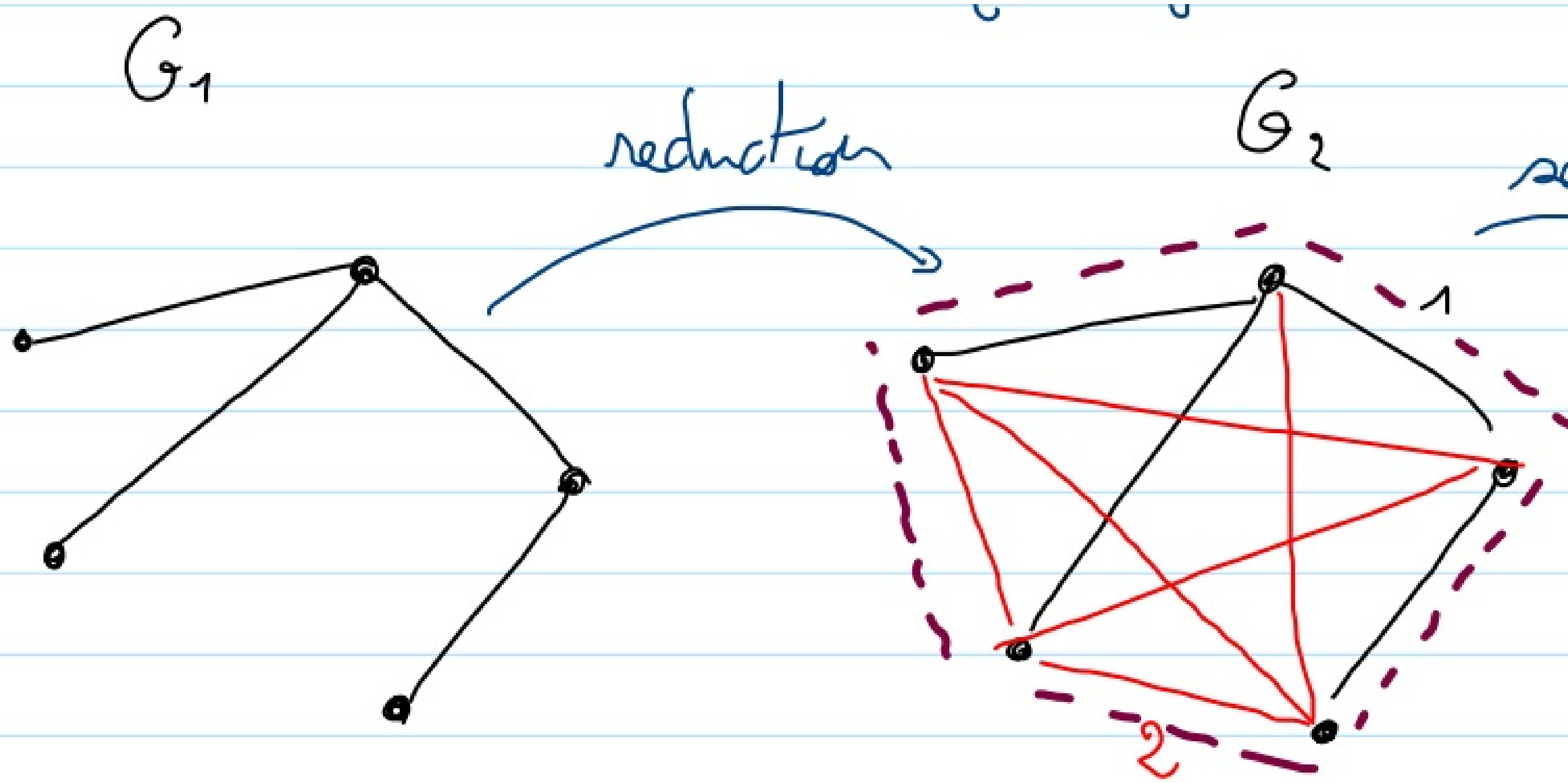


solve D2

$\exists$ TSP tour $\leq W_2$?

cost $= 3 \times 1 + 2 \times 2$
$= 7$

V, & E)



$\exists$ TSP tour $\leq W_2$?

cost $= 3 \times 1 + 2 \times 2$
$= 7$

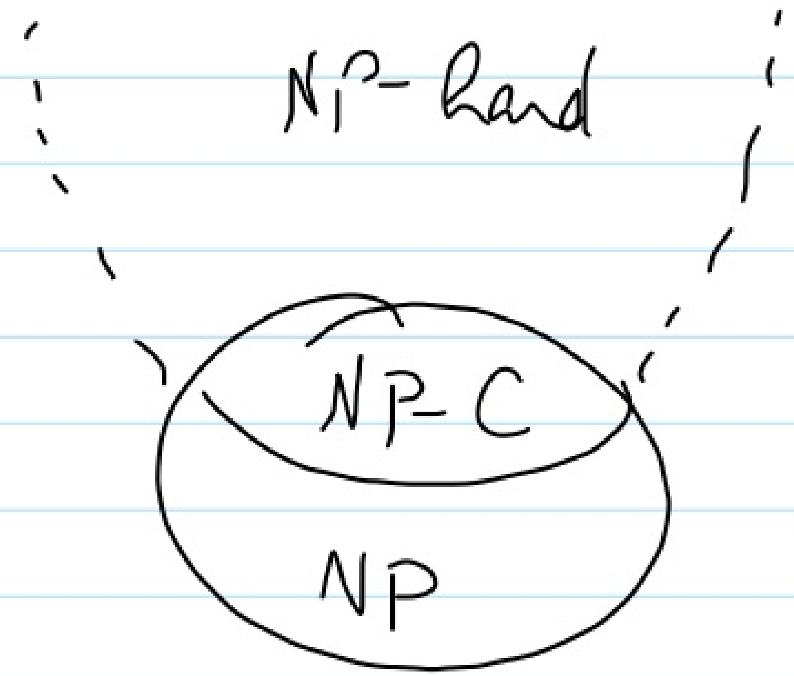
deduce answer to D1

If we choose $W_2 = n$,
then $\exists$ TSP tour $\leq n$ in $G_2 = \varphi(G_1)$ $\iff$ $\exists$ Hamiltonian Circuit in G_1

Therefore, D2 is NP-hard, so P2 is NP-hard as well

Remark: Since we can check the solution of D2 in polynomial time,

$\left.\begin{array}{l} \text{D2} \in \text{NP} \\ \text{D2 is NP-hard} \end{array}\right\}$ D2 is NP-complete



Exercise PL6 (Meunier): Order statistics

$x^{[i]}$ = i-th largest component of vector x

Goal: model $\left(\min \underbrace{\sum_{i=1}^{k} x^{[i]}}_{\text{wtf}} \quad \text{s.t.} \quad \underbrace{\begin{array}{l} Ax = b \\ x \in \mathbb{R}^n_+ \end{array}}_{\text{linear}} \right)$ as a Linear Program

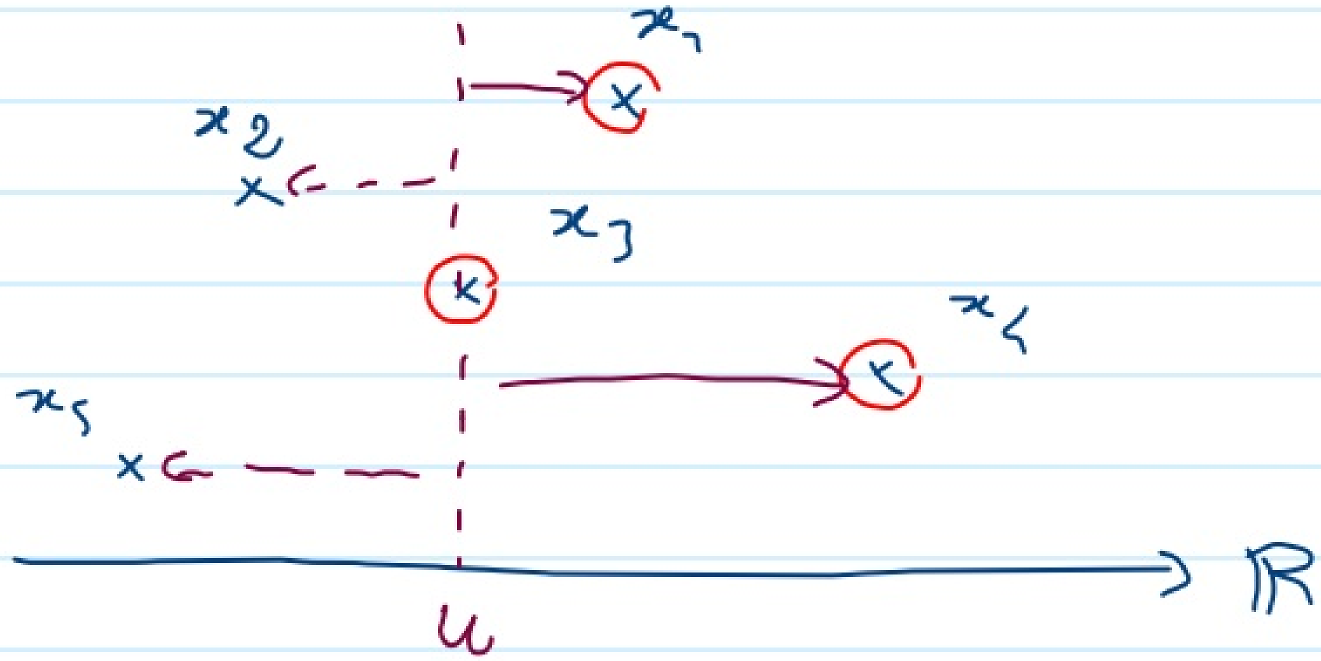
1. $(P) \iff \min_{x} \underbrace{\left(\max_{\substack{S \subset [n] \\ |S| = k}} \sum_{j \in S} x_j \right)}_{\text{wtf}}$ s.t. $Ax = b$, $x \in \mathbb{R}^n_+$

$$\max_{\substack{S \subset [n] \\ |S| = k}} \sum_{j \in S} x_j = \min \; t \quad \text{s.t.} \quad t \geq \sum_{j \in S} x_j \quad \forall S \subset [n], |S| = k$$

$$(P) \iff \min_{x,t} \; t \quad \text{s.t.} \quad \left| \begin{array}{l} t \geq \sum_{j \in S} x_j \quad \underbrace{\forall S \subset [n],\ |S| = k}_{\text{many subsets : } \binom{n}{k}} \\ Ax = b \\ x \in \mathbb{R}^n_+ \\ t \in \mathbb{R} \end{array} \right.$$

$m + n + \binom{n}{k}$ constraints

2.



$k = 3$ $\quad \min \; x^{[1]} + x^{[2]} + x^{[3]}$

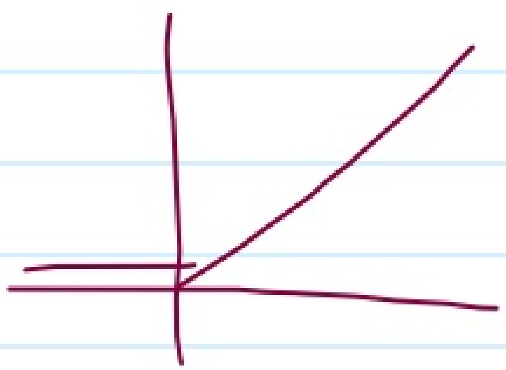
$$x^{[1]} + x^{[2]} + x^{[3]}$$

$$= x_1 + x_3 + x_4$$

$$= 3u + (x_1 - u) + (x_4 - u)$$

$$= 3u + (x_1 - u)^+ + (x_2 - u)^+ + (x_3 - u)^+ + (x_4 - u)^+ + (x_5 - u)^+$$

$$y^+ = \max(y, 0)$$



nonlinear because selection

nonlinear because $y \mapsto y^+$ is not linear

$$(P) \iff \min_{x,u,p} \; ku + \sum_{j=1}^{n} p_j \quad \text{s.t.} \quad \left| \begin{array}{l} p_j \geq x_j - u \\ p_j \geq 0 \\ Ax = b \\ x \geq 0 \end{array} \right.$$

↳ we would need to prove this (the optimal u is $x^{[k]}$)

linear set of constraints

II/ Modeling stuff as Integer Programs (IP)

A Mixed Integer Program is a Linear Program where some variables are forced to take integer values

$$\min_{x \in \underline{\mathbb{Z}^p \times \mathbb{R}^{n-p}}} c^T x \quad \text{s.t. } Ax \leq b \qquad \text{(MILP)}$$

p integer variables
$n-p$ continuous variables

The continuous relaxation of a MILP is obtained by removing the integrality constraints

$$\min_{x \in \underline{\mathbb{R}^n}} c^T x \quad \text{s.t. } Ax \leq b \qquad \text{(R)}$$

Ex: $x \in \mathbb{N}$ becomes $x \geq 0$
$x \in \{0,1\}$ becomes $x \in [0,1]$

less constrained
$\downarrow$
can find solutions with lower costs

Feasible Solutions (MILP) $\subset$ FS(R)

$\Rightarrow$ val (MILP) $\geq$ val (R) lower bound

Complexity :
- (R) is an LP : easy to solve, even in high dim
- (MILP) is hard to solve

↳ solving a MILP requires solving many related LPs (see next class)

Exercise PLM (Meunier) : Column generation

On: how to solve absurdly large LPs
- many constraints : find a way to separate efficiently (spanning tree)
  + constraint / row generation
- many variables : find a way to add a new one efficiently
  + variable / column generation

row of A → constraints | columns → variables

Starting with a subset $I \subset [n]$ of variables, we grow it iteratively until ... when?

(P) min $c^T x$ s.t. $Ax = b$, $x \geq 0$ ⟶ (D)

(P^I) min $c_I^T x_I$ s.t. $A x_I = b$, $x_I \geq 0$ ⟶ (D^I)

duals

Duality switches constraints with variables

constraint in (P) → Lagr. multiplier → var. in (D)

(D^I) has the same variables as (D)

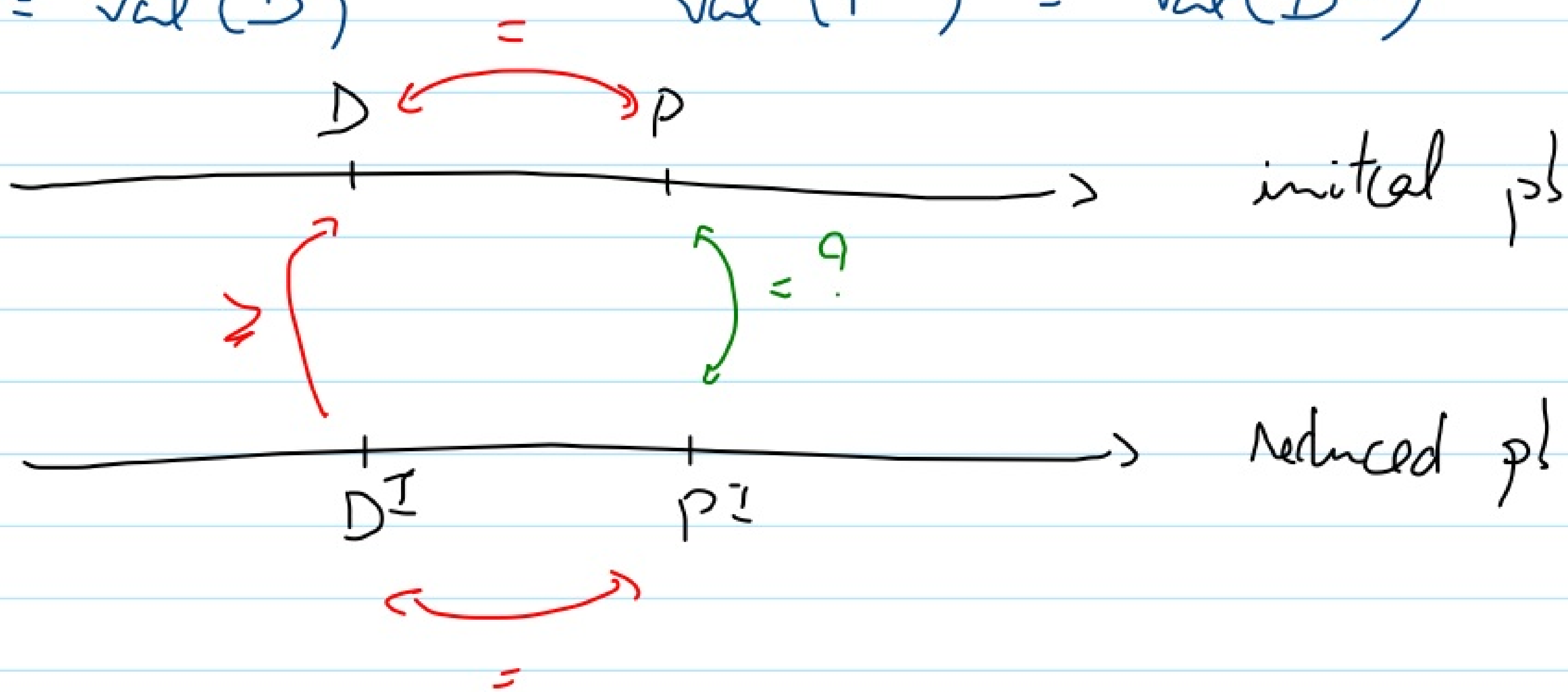
less constraints than (D) because (P^I) has less variables than (P)

$$\mathrm{val}(D^I) \geq \mathrm{val}(D) \qquad \text{(maximization)}$$

There is a solution $\tilde{y}$ to (D^I), also a solution to (D^I)

Strong duality holds for (P) & (D) and (P^I) & (D^I)

$$\mathrm{val}(P) = \mathrm{val}(D) \qquad \mathrm{val}(P^I) = \mathrm{val}(D^I)$$



$$\mathrm{val}(P) = \mathrm{val}(D) \leq \mathrm{val}(D^I) = \mathrm{val}(P^I)$$

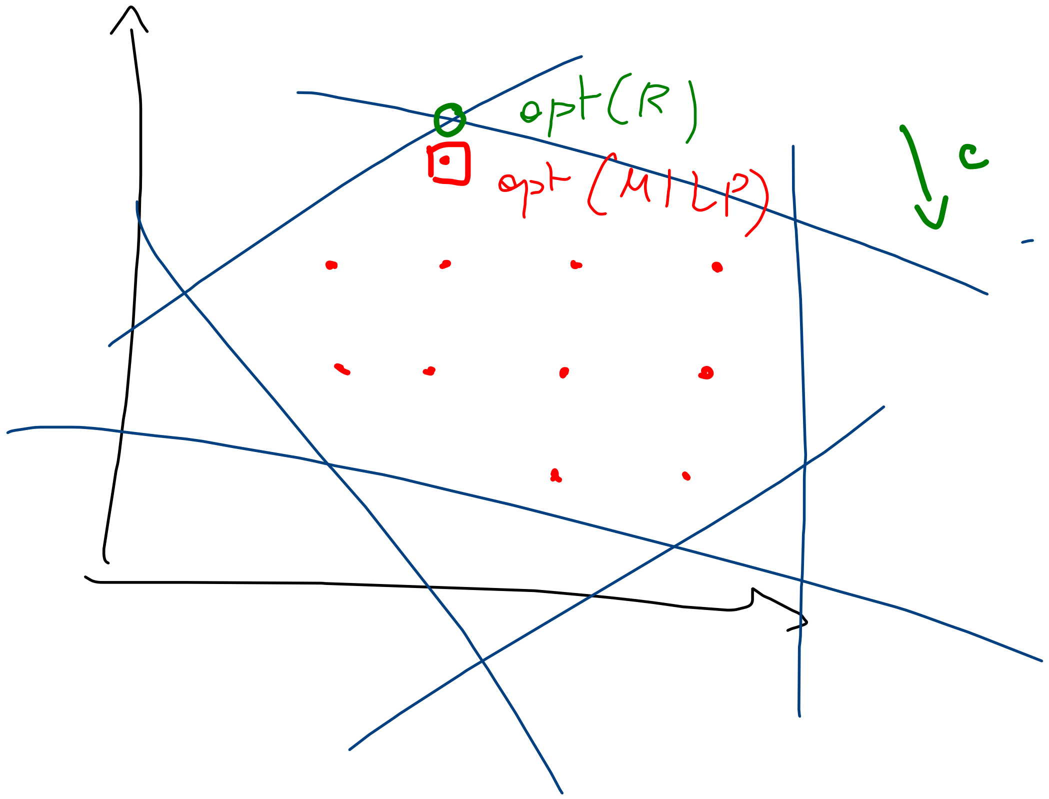
Assumption: $\tilde{y}$ is optimal for (D^I) & feasible for (D)

$\longrightarrow$ $\tilde{y}$ is optimal for (D)

$\mathrm{val}(D^I) = \mathrm{val}(D) = d(\tilde{y})$ so $\mathrm{val}(P) = \mathrm{val}(P^I)$

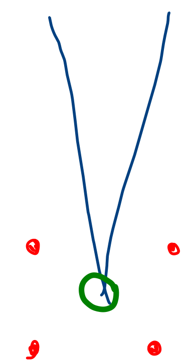
III/ Solving MILPs (ep 1)

Why can't we just solve the relaxation & round to the nearest integer?



This doesn't work (most of the time)

- Sometimes, none of the nearest integer solutions are feasible

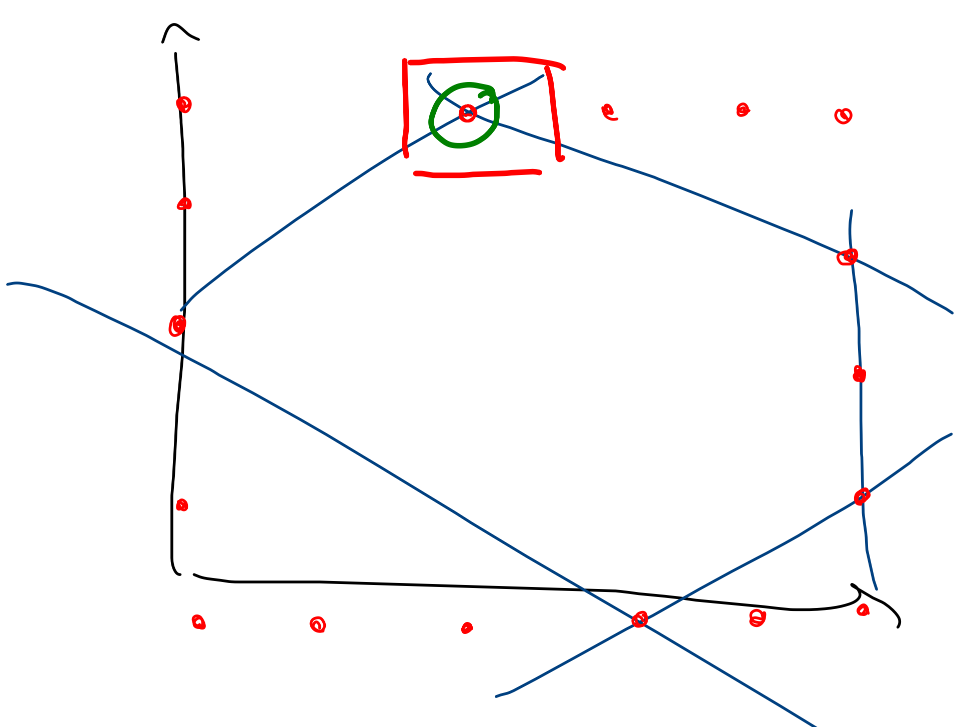


- Rounding is hard in high dimension: 2^d candidates

BUT there is one case where solving (R) is enough:

"integer polyhedra"

↳ those whose vertices all have integer coordinates



Then there is one vertex which is optimal for (R)

& it is also optimal for (MILP)

Criterion: Total Unimodularity (see next class)