# MATH 567: Lecture 1 (01/07/2025)

Integer and Combinatorial Optimization

we will talk mostly about IP (integer programming)

I'm Bala Krishnamoorthy, call me Bala.

Today:
- syllabus, logistics...
- 2D LP example
- convexity, min-max as LP

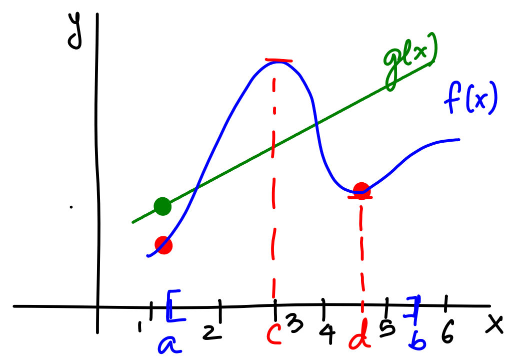
## Optimization in Calculus

$\min f(x)\,,\ x \in [a,b]$

$f'(x) = 0 \rightarrow$ critical points

$f''(x) > 0 \rightarrow$ minima

Also consider end points included in the interval.




$g(x)$: linear, $\min g(x), x \in [a,b] \Rightarrow$ just check end points!

Linear optimization generalizes the special linear 1D case to higher dimensions.

Integer optimization in 1D: $\min f(x)$, $x \in [a,b]$, $x \in \mathbb{Z}$

insist on finding integer optimal solutions

$\mathbb{Z}$: set of integers

A standard linear program (LP)

$$\begin{aligned} \max \quad & \bar{c}^T \bar{x} \\ \text{s.t.} \quad & A\bar{x} \le \bar{b} \\ & \bar{x} \in \mathbb{R}^n_{\ge 0} \end{aligned}$$

$\bar{c}, \bar{x} \rightarrow$ vectors (lower-case letters w/ bar)

my notation in these notes!

or $=$ (as used in many books)

Here is an example of LP formulation, and graphical solution.

(Taken from *Introduction to Mathematical Programming* by Winston and Venkataramanan.)

---

Farmer Jones must decide how many acres of corn and wheat to plant this year. An acre of wheat yields 25 bushels of wheat and requires 10 hours of labor per week. An acre of corn yields 10 bushels of corn and requires 4 hours of labor per week. Wheat can be sold at \$4 per bushel, and corn at \$3 per bushel. Seven acres of land and 40 hours of labor per week are available. Government regulations require that at least 30 bushels of corn need to be produced in each week. Formulate and solve an LP which maximizes the total revenue that Farmer Jones makes.

---

Decision variables (d.v.s):

$x_i$ = # acres of crop $i$, $i = 1, 2$ (1 = corn, 2 = wheat)

$$\text{maximize} \quad z = 3\cdot 10\cdot x_1 + 4\cdot 25\cdot x_2 \qquad \text{(total revenue)}$$

s.t.

subject to

$$x_1 + x_2 \leq 7 \qquad \text{(total land)}$$

$$4x_1 + 10x_2 \leq 40 \qquad \text{(total labor hrs)}$$

$$10x_1 \geq 30 \qquad \text{(min corn)}$$

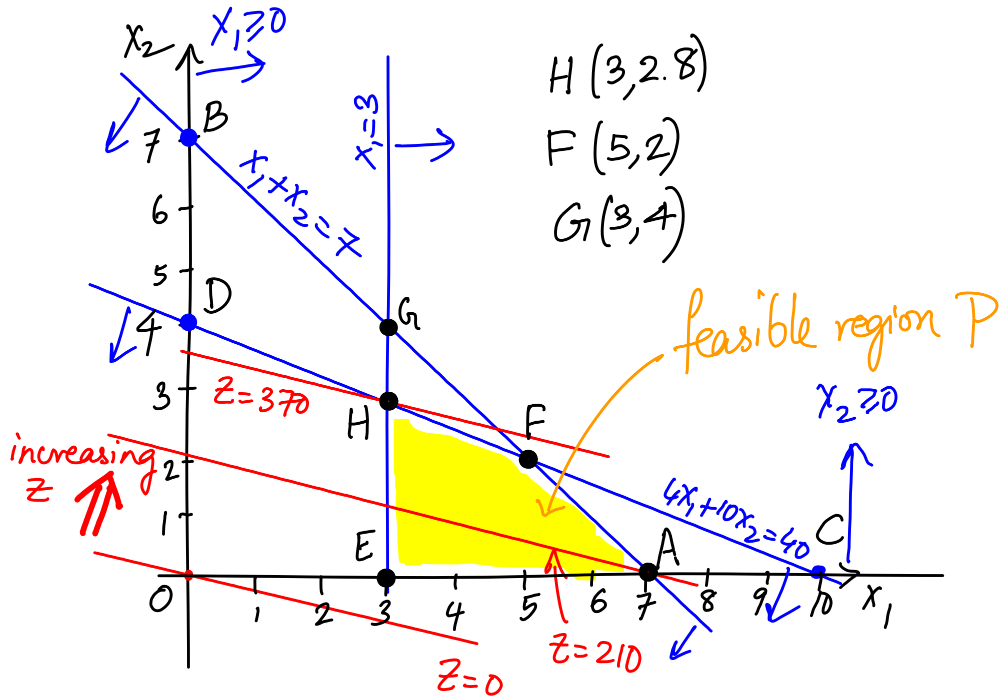
$$x_1, x_2 \geq 0 \qquad \text{(non-neg)}$$

linear program (LP)

The above model is an LP formulation. You need to specify the d.v.s, and describe the objective function and constraints as linear functions or (in)equalities in the d.v.'s.

# Graphical Solution of the LP

slide z-line up (in the direction of increasing z)




Optimal solution is at H (3, 2.8), with optimal $z^* = 370$.

## Points to note

* Optimal solution (if one exists) occurs at a corner point (vertex)
* the feasible region P is convex

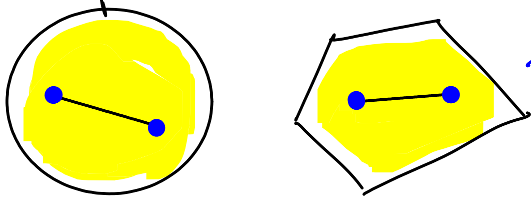
The first result depends on the fact that P is convex — indeed, we will strive hard to utilize convexity whenever possible!

**Def** P is convex if $\forall$ $0 \leq \lambda \leq 1$, $\bar{x}_1, \bar{x}_2 \in P$,

$$\lambda \bar{x}_1 + (1-\lambda) x_2 \in P.$$

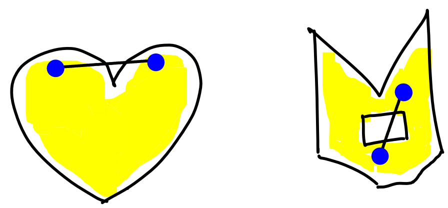
In words, the line segment connecting $\bar{x}_1$ and $\bar{x}_2$ lies in P.

Examples of convex regions:



We will talk a lot about polytopes and polyhedra

not convex!

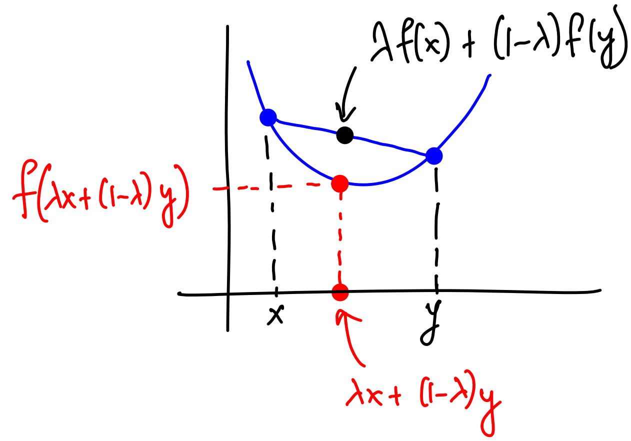


**Convex function** $f: \mathbb{R}^n \to \mathbb{R}$, is **convex** if

$\forall\, \bar{x}, \bar{y} \in \mathbb{R}^n$, $\lambda \in [0,1]$, we have

$$f(\lambda \bar{x} + (1-\lambda)\bar{y}) \leq \lambda f(\bar{x}) + (1-\lambda) f(\bar{y})$$

$\geq$ → concave




Again, convexity plays a critical role in how we formulate LPs and integer LPs. There are certain scenarios which could be modeled as LPs due to convexity, but other that cannot be!

**Theorem**: Let $f_1, \dots, f_m : \mathbb{R}^n \to \mathbb{R}$ be convex. Then

$$f(\bar{x}) = \max_{i=1,\dots,m} f_i(\bar{x}) \text{ is convex.}$$

We will stick to the finite cases in this class, i.e., m and n in the above theorem are finite.

**Proof** Let $\bar{x}, \bar{y} \in \mathbb{R}^n$, $\lambda \in [0, 1]$.

$$f(\lambda\bar{x} + (1-\lambda)\bar{y}) = \max_{1 \le i \le m} f_i(\lambda\bar{x} + (1-\lambda)\bar{y})$$

$$\le \max_{1 \le i \le m} \lambda f_i(\bar{x}) + (1-\lambda) f_i(\bar{y})$$ as each $f_i$ is convex

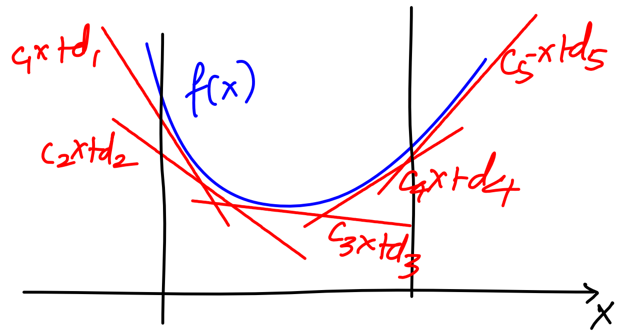
$$\le \lambda \underbrace{\left[\max_{1 \le i \le m} f_i(\bar{x})\right]}_{f(\bar{x})} + (1-\lambda) \underbrace{\left[\max_{1 \le i \le m} f_i(\bar{y})\right]}_{f(\bar{y})}$$

$$= \lambda f(\bar{x}) + (1-\lambda) f(\bar{y}) \qquad \square$$

**Def** $f(\bar{x}) = \max_{1 \le i \le m} (\bar{c}_i^T \bar{x} + d_i)$, $\bar{c}_i \in \mathbb{R}^n$, $d_i \in \mathbb{R}$ is a **piecewise linear (PL) convex function**, with convexity following from the previous result.

A (general) convex function could be approximated efficiently using a PL convex function in certain optimization problems.

The more number of linear pieces we use, the finer the approximation of $f(x)$ is.




Consider the following generalization of LP:

$$\begin{aligned} \min \quad & \max_{1\le i\le m} (\bar{c}_i^T\bar{x}+d_i) \\ \text{s.t.} \quad & A\bar{x}\le\bar{b} \\ & \bar{x}\geq\bar{0} \end{aligned}$$

not an LP as written (as the objective is not $\bar{c}^T\bar{x}$)

We can write an equivalent LP with an extra variable and $m$ extra constraints:

$$\text{LP}\left\{\begin{aligned} \min \quad & z \\ \text{s.t.} \quad & z\geq \bar{c}_i^T\bar{x}+d_i, \quad i=1,\dots,m \\ & A\bar{x}\le\bar{b} \\ & \bar{x}\geq\bar{0} \end{aligned}\right.$$

Note: We are able to model as an LP because the objective function is a min-max one.

What if it were a min-min or a max-max one?

We will have to use integer variables!

$$\left\{\begin{aligned} & \max\ z \\ & z\geq\bar{c}_i^T\bar{x}+d_i \\ & A\bar{x}\le\bar{b} \\ & \bar{x}\geq\bar{0} \end{aligned}\right.$$

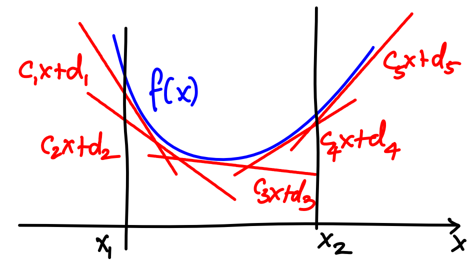
this LP is unbounded!

Analogy:

$z$ models a "blanket" that stays above all the lines (linear "pieces") $\Leftarrow$ $z \geq \bar{c}_i^T \bar{x} + d_i$, $i = 1, \ldots, m$.

Minimizing $z$ pushes the blanket plush against the pieces from above, and hence $z$ models the function as desired.

If we maximize $z$ instead, the blanket is pulled up, and there is no limit on how far it can be pulled up. The problem is unbounded in this case!



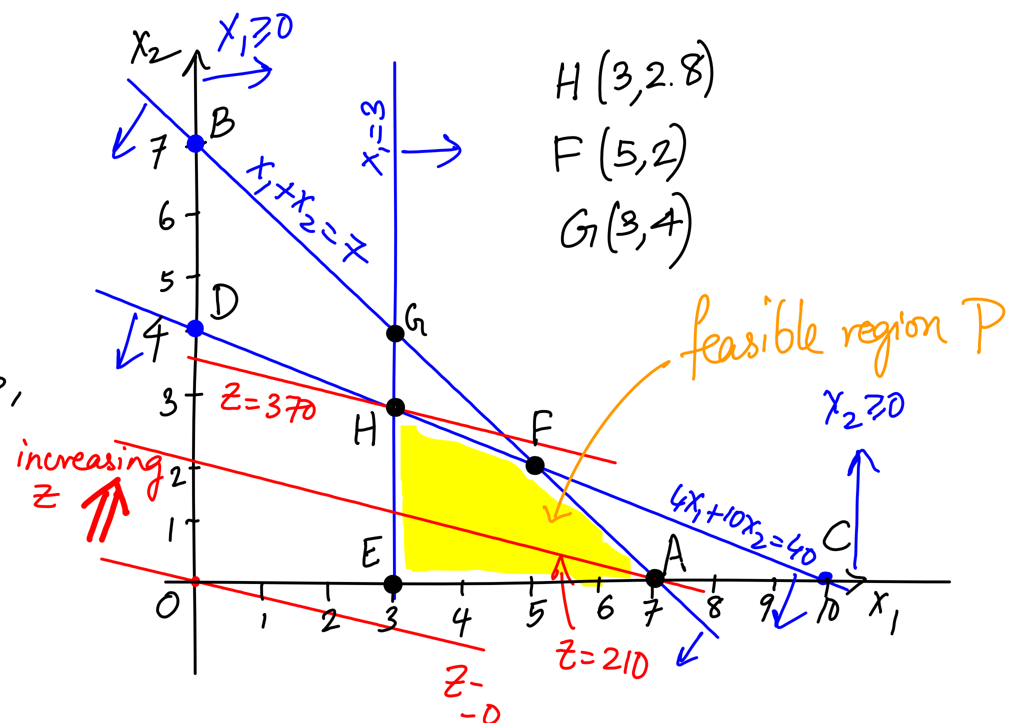

The first instinct on given an IP is to drop the integer restrictions and solve the underlying LP. May be then we can "round" the LP solution to a nearest integral solution to get a solution for the IP. But this idea fails in many cases.

Farmer Jones LP:

$H(3, 2.8)$ is the optimal solution.

If $x_1, x_2 \in \mathbb{Z}$ on top, rounding H will put you outside P!

$(3,3) \notin P$




Rounding down gives $(3,2)$, which is feasible. But the optimal integer solution is $F(5,2)$.

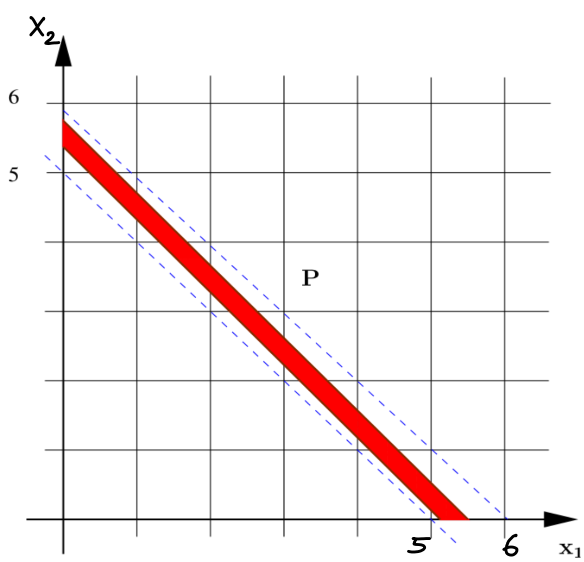
# MATH 567: Lecture 2 (01/14/2025)

Today: * general forms of IP
* IP formulations

To round off (no pun intended!) the discussion on possibly rounding fractional solutions from LP relaxations to solve integer programs, we present another "extreme" example.

Consider the polytope $P$ shown here, which is defined by the following constraints.

$$\left.\begin{cases} 106 \leq 21x_1 + 19x_2 \leq 113 \\ 0 \leq x_1, x_2 \leq 6 \end{cases}\right\} (P)$$




As is evident from the figure, $P \cap \mathbb{Z}^2 = \emptyset$! The idea of rounding for any IP defined on $P$ is moot here.

for any objective function

We will return to such examples later on.

# General Forms of integer (linear) Programs

## Mixed integer Program (MIP)

$$\begin{aligned} \max \quad & z = \bar{c}^T \bar{x} + \bar{d}^T \bar{y} \\ \text{s.t.} \quad & A\bar{x} + B\bar{y} \leq \bar{b} \\ & \bar{x} \in \mathbb{Z}_{\geq 0}^{n_1}, \; \bar{y} \in \mathbb{R}_{\geq 0}^{n_2} \end{aligned} \qquad \text{(MIP)}$$

## Pure integer program (IP)

$$\begin{aligned} \max \quad & z = \bar{c}^T \bar{x} \\ \text{s.t.} \quad & A\bar{x} \leq \bar{b} \\ & \bar{x} \in \mathbb{Z}_{\geq 0}^{n} \end{aligned} \qquad \text{(IP)}$$

## Special case of IP: Binary IP (BIP)

$$\begin{aligned} \max \quad & z = \bar{c}^T \bar{x} \\ \text{s.t.} \quad & A\bar{x} \leq \bar{b} \\ & \bar{x} \in \{0,1\}^{n} \end{aligned} \qquad \text{(BIP)}$$

Q When do we insist $x_i \in \mathbb{Z}$?

Should we build a new dorm? $\Rightarrow$ $x \in \{0,1\}$ insist on it.

fractional value does not make sense

How many rooms should we build?

$$y = 385.6 \begin{cases} \nearrow 386 \\ \searrow 385 \end{cases}$$ both are okay in the big scheme of things.

So, just set $y \in \mathbb{R}_{\geq 0}$.

We will now look at several BIP and MIP formulation problems. It is important to remember that one **need not write these formulations in standard form**. Later on, when we describe algorithms to solve these problem instances, it will make sense to describe them for problems in standard form. Similarly, when we study the geometry or other properties of the associated polytope, we will do so using a standard form. In fact, it is better to write formulations in non-standard form if they are more readable!

We will introduce AMPL, which is a state-of-the-art **modeling software**. The function of such a software is to convert formulations written in non-standard form to, sometimes more concise, standard form. Then AMPL sends the standard form problem to a solver, e.g., Gurobi, which runs linear-algebra based algorithms to solve the same. AMPL then "interprets" the solution to the standard form problem back to the original form before displaying the same.

We will first introduce several instances of the BIP, and then the MIP. Later on, we will describe some unified procedures to model most situations using only binary variables, or using binary and continuous variables. Further on, we will discuss how to "compare" formulations – as it turns out, there are multiple ways to formulate the same situation, and one formulation might be "tighter" than the rest.

# BIP formulations

## 1. Assignment Problem

$n$ persons, $n$ jobs, $c_{ij}$ = cost of person $i$ doing job $j$

Goal: Assign each person to a job so that total cost is minimum.

Step 1 decision variables (d.v.s)

Let $x_{ij} = \begin{cases} 1 & \text{if person } i \text{ does job } j, \\ 0 & \text{o.w.} \end{cases}$ $n^2$ vars.

o.w. → "otherwise"

Step 2: Constraints

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall i \ (i=1,\dots,n) \quad \text{(person } i \text{ gets one job)}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j \ (j=1,\dots,n) \quad \text{(job } j \text{ gets one person)}$$

$$x_{ij} \in \{0,1\} \quad \forall i,j$$

Step 3 Objective function

$$\min \ z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \quad \text{(total cost)}$$

In summary

$$\min \ z = \sum_i\sum_j c_{ij}x_{ij} \quad \text{(total cost)}$$

s.t.

$$\sum_j x_{ij} = 1 \ \forall i \quad \text{(person } i \text{ gets 1 job)}$$

$$\sum_i x_{ij} = 1 \ \forall j \quad \text{(job } j \text{ gets 1 person)}$$

$$x_{ij} \in \{0,1\} \quad \forall i,j.$$

As you do more formulations, you will naturally go straight to the compact form, rather than write out the detailed steps. But do define the d.v.'s first in all cases!

## 2. The 0-1 knapsack problem

* $n$ projects.
* total budget $b$.
* cost of project $j$ is $a_j$.
* value of project $j$ is $c_j$.

Decide which projects to choose within the budget such that total value is maximized.

It is assumed that you cannot undertake a fraction, e.g., 0.4., of a project.

d.v's $x_j = 1$ if project $j$ is selected, 0 o.w.

Constraints $\sum_{j=1}^{n} a_j x_j \le b$ (budget)

$x_j \in \{0,1\} \quad \forall j$ (binary vars).

objective function $\max \sum_{j=1}^{n} c_j x_j$.

$$\max \sum c_j x_j$$
$$\text{s.t.} \quad \sum a_j x_j \le b$$
$$x_j \in \{0,1\} \quad \forall j.$$

equality knapsack problem: $=$ have to use up all the available budget

In feasibility knapsack, we want to find $x_j \in \{0,1\}$ s.t. $\sum a_j x_j = b$. (or, more generally, $\{b' \le \bar{a}^T \bar{x} \le b\}$)

$\bar{a}^T \bar{x} = \sum a_j x_j$

There is no objective function specified.

## Additional Restrictions

1. If project 2 is chosen, so must be project 5.

$$x_5 \geq x_2$$

If $x_2 = 1$, then we have $x_5 \geq 1$, which forces $x_5 = 1$, as $x_5 \in \{0,1\}$.
But if $x_2 = 0$, we get $x_5 \geq 0$, which is redundant.

Notice that the reverse implication that "if project 5 is chosen, then so must project 2" is not forced by this constraint. Indeed, if $x_5 = 1$, we get $x_2 \leq 1$, which is redundant.

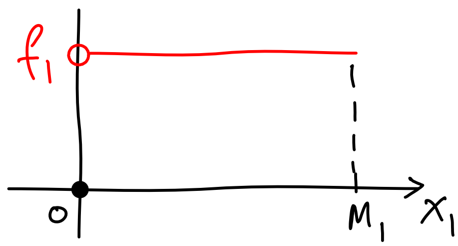
Note that $x_2 = x_5$ is also not correct here, as that constraint models "either pick both projects 2 and 5, or neither one."

2. Must ~~Can~~ choose exactly ~~(at most)~~ at least 2 out of projects 1, 3, 6, 7.

$$x_1 + x_3 + x_6 + x_7 \; \overset{=}{\underset{\geq}{\leq}} \; 2$$

It is important to realize that these constraints work as intended only when all $x_j$'s are binary. Further, we do not want to force more than what is required. We will first try to write the required constraints just using logic, but will later describe a more systematic approach.

# 3. Fixed Charge (continued...)




$$\min \ f(x_1) + c_2x_2 + \cdots + c_nx_n$$
$$A\bar{x} \le \bar{b}$$
$$0 \le x_1 \le M_1$$

$$f(x_1) = \begin{cases} 0, & x_1 = 0 \\ f_1, & x_1 > 0 \end{cases} \quad (f_1 > 0).$$

YES/NO question: Is $x_1 > 0$?

$\Longrightarrow$ model using $y_1 \in \{0,1\}$.

We want $y_1 = \begin{cases} 1 & \text{if } x_1 > 0 \\ 0 & \text{o.w.} \end{cases}$

$$\boxed{\begin{array}{ll} \min & f_1y_1 + c_2x_2 + \cdots + c_nx_n \\ \text{s.t.} & A\bar{x} \le \bar{b} \\ & 0 \le x_1 \le M_1y_1 \\ & y_1 \in \{0,1\} \end{array}}$$

If $x_1 > 0$, $y_1$ is forced to 1. Else, $x_1 \le My_1$ will not hold (with $y_1 = 0$).

If $x_1 = 0$, $x_1 \le M_1y_1$ can hold with $y_1 = 0$ or $y = 1$. But the term $f_1y_1$ in the min objective function forces $y_1 = 0$ in the optimal solution. Recall, $f_1 > 0$ here.

## 4. Interactive fixed charge → Included in HW1!

Similar to Problem 3 but

$$f(x_1, x_2) = \begin{cases} 0, & \text{if } x_1 = x_2 = 0 \\ f_1, & \text{if } x_1 > 0,\ x_2 = 0 \\ f_2, & \text{if } x_1 = 0,\ x_2 > 0 \\ f_{12}, & \text{if } x_1 > 0 \ \&\ x_2 > 0 \end{cases}$$

with $0 \le x_1 \le M_1$, $0 \le x_2 \le M_2$, $f_{12} \ge f_1 > 0$, $f_{12} \ge f_2 > 0$, $f_{12} \ne f_1 + f_2$

need not hold as equality; indeed $f_{12} \gtreqless f_1 + f_2$ are all ok.

Can we use $y_1, y_2 \in \{0,1\}$, and $y_1 \times y_2$? Yes, but $y_1 y_2$ is nonlinear, so you want to somehow linearize it — maybe define $y_{12} \in \{0,1\}$, and relate it to $y_1$ and $y_2$ using extra constraints.

Later on, we will talk about linearizing such nonlinear terms - products of binary variables, or $x_i^2$ when $x_i \in \{0,1\}$, etc.

# MATH 567 : Lecture 3 (01/16/2025)

Today : * More MIP formulations
* modeling tools for BIPs

---

Recall : min-max objective functions and constraints — could be modeled as linear programs.

e.g., $\min\{|x|\} \longrightarrow \min\{\max\{x, -x\}\}$

$\longrightarrow \min\{z \mid z \geq x,\ z \geq -x\}$.

Similarly, we could model $\max\{\cdot,\cdot\} \leq b$ or $\min\{\cdot,\cdot\} \geq b$ constraints as equivalent linear systems. For instance,

$$|x| \leq 5 \longrightarrow \max\{x, -x\} \leq 5 \longrightarrow x \leq 5,\ -x \leq 5.$$

But $|x| \geq 4$ cannot be modeled as an LP. In particular, ~~$x \geq 4$ and $-x \geq 4$~~ is not what we want.

Will have to use an extra binary variable to model which of two options holds in this case.

Recall : Fixed charge :

$$\min f_1 y_1 + \cdots \qquad (f_i > 0)$$

s.t. $\ldots$

$$x_1 \leq M_1 y_1 \qquad y_1 \in \{0,1\}$$

We will see another problem class where fixed charge shows up. Later, we will see how to force the relation between $x_1$ and $y_1$ without relying on the min $f_1 y_1$ objective function.

# 5. Uncapacitated lot sizing (ULS)

* 1 product, $n$ time periods $(t = 1, \dots, n)$
* $d_1, \dots, d_n$ : demand in each time period
* $f_1, \dots, f_n$ : fixed cost for making any $>0$ # items in each time period
* $c_1, \dots, c_n$ : unit production cost in each time period
* $h_1, \dots, h_n$ : unit holding (or storage) costs ($h_t$ : cost for storing one unit from period $t$ to $t+1$)

Goal: production plan that minimizes total cost.

Assumptions:
* infinite production capacity (no storage capacity as well)
* no units to start with, or at end

d.v.s :

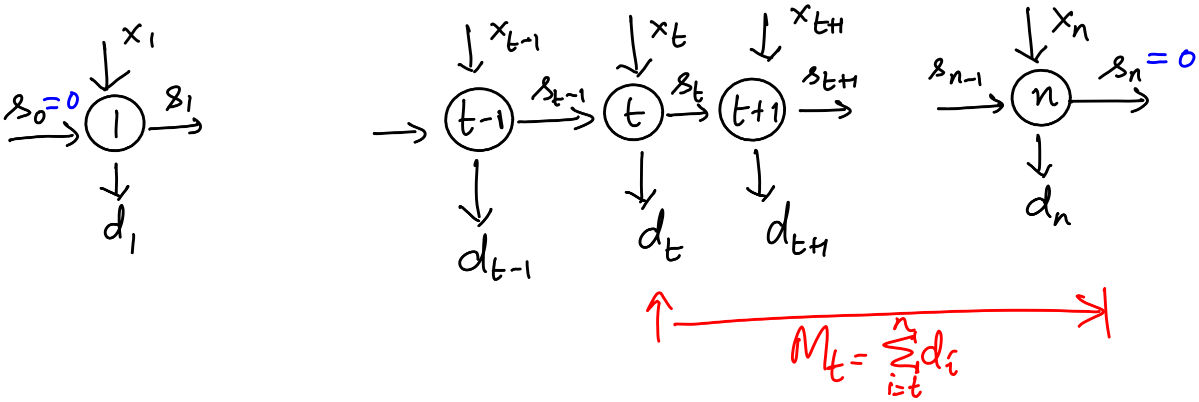
$x_t$ = # units produced in period $t$, $t = 1, \dots, n$ ($\geq 0$, continuous)

$s_t$ = # units stored from period $t$ to $t+1$, $t = 0, \dots, n$ ($\geq 0$, continuous)

$$y_t = \begin{cases} 1 & \text{if } x_t > 0 \\ 0 & \text{o.w.} \end{cases}, \quad t = 1, \dots, n$$

→ to capture the fixed charge terms

Here is a schematic:




$d_t$'s are data given to us

Here is the MIP: → we do have an MIP, as $s_t, x_t$ are continuous, while $y_t$ is binary

$$\min \sum_{t=1}^{n} f_t y_t + \sum_{t=1}^{n} c_t x_t + \sum_{t=1}^{n} h_t s_t \qquad \text{(total cost)}$$

s.t.

$$s_0 = 0, \quad s_n = 0 \qquad \text{(no start/end inventory)}$$

$$\underbrace{s_{t-1} + x_t}_{\text{inflow}} = \underbrace{d_t + s_t}_{\text{outflow}}, \quad t = 1, \ldots, n \qquad \text{(flow balance)}$$

$$x_t \le M_t y_t \quad t = 1, \ldots, n \qquad \text{(forcing constraints)}$$

$$s_t \ge 0, x_t \ge 0, \; y_t \in \{0,1\} \; \forall t \qquad \text{(var. restrictions)}.$$

What should $M_t$ be? Any large enough ($>0$) number will work, But ideally, use the smallest $M_t$ that works.

$$M_t = \sum_{i=t}^{n} d_i \quad \text{will work here.}$$

We will spend a lot of time on details such as the choice of $M_t$, and how they affect the "strength" of the formulation.

If we allow backlogging, demand in period $t$ could be satisfied by (part of) $x_j$ for $j > t$. In this case,

$$M_t = \sum_{i=1}^{n} d_i \quad \text{will work,}$$

since all the demand could potentially be satisfied by producing in the same single period.

## 6. (General) Piecewise linear function (not convex in the interesting case)

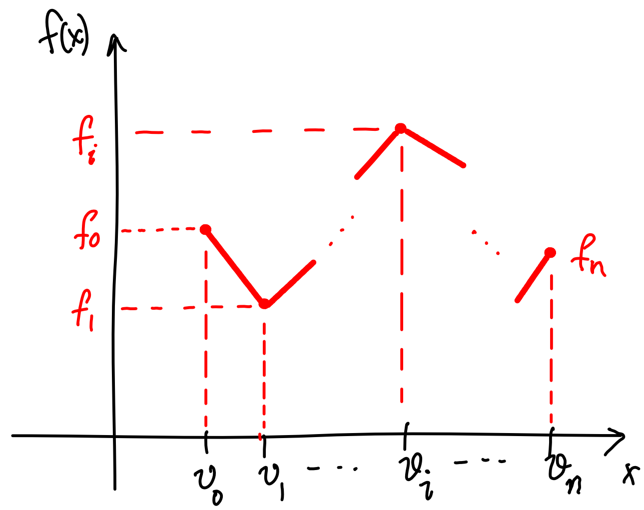
$x$ : scalar

$$f(x) = \begin{cases} f_i, & \text{if } x = v_i \quad (i = 0, \dots, n) \\ \text{linear}, & \text{if } v_i \le x \le v_{i+1} \quad (i = 0, \dots, n-1) \end{cases}$$

If $x = \lambda_i v_i + \lambda_{i+1} v_{i+1}$

$\lambda_i, \lambda_{i+1} \geq 0$, $\lambda_i + \lambda_{i+1} = 1$, then

$f(x) = \lambda_i f_i + \lambda_{i+1} f_{i+1}$




Let $\delta_i = v_i - v_{i-1}$. We let

$s_i = \dfrac{f_i - f_{i-1}}{\delta_i}$, $i = 1, \dots, n$ (slopes, can be $\geq 0$ or $\leq 0$).

Let $x_i$ be "the portion of $x$ in $[v_{i-1}, v_i]$", $i = 1, \dots, n$.

If we

1. write $x = v_0 + \sum_{i=1}^{n} x_i$

$$g = f_0 + \sum_{i=1}^{n} s_i x_i$$

$$0 \le x_i \le \delta_i,$$

2. somehow enforce

"if $x_{i+1} > 0$ then $x_i \geq \delta_i$", for $i = 1, \dots, n-1$

3. plug in $g$ for $f(x)$;

then we're done!

3.5

$A \Rightarrow B \equiv$
$\neg A$ or $B$
"not"

2. Equivalent logical expression:

"either $x_{i+1} \le 0$ or $x_i \ge \delta_i$"

$-x_i + \delta_i \le 0$

Let $y_i$ and $z_i$ are 0-1 variables

$$\boxed{\begin{aligned} x_{i+1} &\le \delta_{i+1} y_i \\ -x_i + \delta_i &\le \delta_i z_i \\ y_i + z_i &= 1 \\ y_i, z_i &\in \{0,1\} \end{aligned}}$$

if $x_{i+1} > 0$ then $y_i = 1$ $\Rightarrow z_i = 0$

$\forall i = 1, \ldots, n-1$

assuming XOR "exclusive OR"
A or B, but not both

If $x_{i+1} > 0$, then $y_i = 1 \Rightarrow z_i = 0$ (as $y_i + z_i = 1$).

$\Rightarrow -x_i + \delta_i \le 0 \Rightarrow x_i \ge \delta_i$

Can simplify:

$$x_{i+1} \le \delta_{i+1} y_i$$

$$-x_i + \delta_i \le \delta_i (1 - y_i)$$

as $y_i + z_i = 1$

$$x_i \ge \delta_i y_i$$

i.e., $\delta_i y_i \le x_i \le \delta_i y_{i-1}$, $i = 1, \ldots, n-1$

$x_{i+1} > 0 \Rightarrow y_i = 1 \Rightarrow y_{i-1} = 1, y_{i-2} = 1, \ldots, y_1 = 1$.

So, we can force both implications for $y_i = \begin{Bmatrix} 1, \text{ if } x_i > 0 \\ 0 \text{ o.w} \end{Bmatrix}$ using constraints, i.e., do not have to rely on a min $f_i y_i$ objective function.

We present one last formulation instance...

## 7. Semicontinuous variable

Need that "x does not take values that are too small".

e.g., if you buy any of a stock option, you need to buy at least 100 of them.

statement: $x$ is zero or is at least $\ell$ $(>0)$ (and $\leq M$)

Model: $\boxed{\ell y \leq x \leq My, \; y \in \{0,1\}}$

We now consider some themes/governing principles for writing all such formulations.

## 1. Modeling with only 0-1 variables

$x_1, x_2, \ldots$ are 0-1 (binary) variables

### Notation

$L_i \equiv (x_i = 1)$

$\vee \equiv$ OR, $\wedge \equiv$ AND

$\Rightarrow \equiv$ "implies", $\Leftrightarrow \equiv$ "equivalent"

$\neg \equiv$ NOT (negation)

These are standard notation used in mathematical logic. We will start with statements, and then try to write the model, i.e., set of inequalities, that represents the statement.

# Examples

| statement | model (constraints) |
|---|---|
| 1. $L_1 \vee L_2 \vee \cdots \vee L_n$ | $x_1 + x_2 + \cdots + x_n \geq 1$ |
| 2. $L_1 \Rightarrow L_2$ | $x_1 \leq x_2$ |
| 3. $L_1 \Leftrightarrow (L_2 \wedge L_3)$ | |

i.e.,

$$\begin{cases} L_1 \Rightarrow (L_2 \wedge L_3) \\ L_1 \Leftarrow (L_2 \wedge L_3) \end{cases}$$

$L_1 \Rightarrow (L_2 \wedge L_3)$: $x_1 \leq x_2,\ x_1 \leq x_3$

$L_1 \Leftarrow (L_2 \wedge L_3)$: think about it! We'll present the model in the next lecture...

# MATH 567: Lecture 4 (01/21/2025)

Today:
* conjunctive normal form (CNF)
* model with 0-1 and continuous variables
* arbitrary disjunctions

## Modeling with 0-1 variables (continued...)

3. $L_1 \Leftrightarrow (L_2 \wedge L_3)$

i.e.,

$$\begin{cases} L_1 \Rightarrow (L_2 \wedge L_3) & \to\ x_1 \le x_2,\ x_1 \le x_3 \ \text{ OR } \ 2x_1 \le x_2 + x_3 \\ L_1 \Leftarrow (L_2 \wedge L_3) & \to\ \cancel{x_1 \ge \frac{x_2+x_3}{2}} \end{cases}$$

$\cancel{x_1 \ge \frac{x_2+x_3}{2}}$ will force $x_1 = 1$ when $x_2 = 1, x_3 = 0$!

$\cancel{x_1 \ge x_2 x_3}$ nonlinear!

$\cancel{2x_1 = x_2 + x_3}$ $x_2 = x_3 = 0$ forces $x_1 = 0$!

$x_1 \ge x_2 + x_3 - 1$

**Q.** Is there a general method to model any logical statement?

YES! As long as the statement is in a "nice" form.
And every statement has such a "nice" form!

**Def** A **literal** is an elementary statement, e.g., $L_i$, $\neg L_j$.

A **clause** is a set of literals connected with "OR" ($\vee$)

e.g., $L_1 \vee L_3$, $\neg L_2 \vee L_4 \vee \neg L_5$.

**Def** A logical statement is in **conjunctive normal form** (CNF) if it is a set of clauses connected by ANDs ($\wedge$).

e.g., $(L_1 \vee L_3) \wedge (\neg L_2 \vee L_3 \vee \neg L_5) \wedge (\neg L_3 \vee L_7)$

is in CNF.

If a statement is in CNF, it is easy to write down its representative model using inequalities.

e.g., $\left\{ \begin{array}{r} x_1 + x_3 \geq 1 \\ (1-x_2) + x_3 + (1-x_5) \geq 1 \\ (1-x_3) + x_7 \geq 1 \end{array} \right\}$ is a model for the statement in CNF above.

**Claim** Every (finite) statement involving $\vee, \wedge, \neg, \Rightarrow, \Leftrightarrow$ has a CNF. The CNF may not be unique.

## Some Rules for doing transformations

① $L_1 \wedge (L_2 \vee L_3) \equiv (L_1 \wedge L_2) \vee (L_1 \wedge L_3)$

② $L_1 \vee (L_2 \wedge L_3) \equiv (L_1 \vee L_2) \wedge (L_1 \vee L_3)$

③ $\neg (L_1 \wedge L_2) \equiv \neg L_1 \vee \neg L_2$

④ $\neg (L_1 \vee L_2) \equiv \neg L_1 \wedge \neg L_2$

⑤ $L_1 \Rightarrow L_2 \overset{\text{def}}{\equiv} \neg L_1 \vee L_2$

We could replace literals with clauses, or more general statements in the above rules, and they still hold, e.g., $C_1 \Rightarrow C_2 \equiv \neg C_1 \vee C_2$.

clauses

## Examples

1. $(L_2 \wedge \cdots \wedge L_n) \Rightarrow L_1 \equiv \neg(L_2 \wedge \cdots \wedge L_n) \vee L_1$

$$\equiv (\neg L_2 \vee \neg L_3 \vee \cdots \vee \neg L_n) \vee L_1$$

$$\equiv \neg L_2 \vee \neg L_3 \vee \cdots \vee \neg L_n \vee L_1$$

which is in CNF.

model: $(1-x_2)+(1-x_3)+\cdots+(1-x_n)+x_1 \geq 1$

2. $(L_1 \wedge L_2) \vee (L_3 \wedge (L_4 \vee L_5))$

$$\equiv ((L_1 \wedge L_2) \vee L_3) \wedge ((L_1 \wedge L_2) \vee (L_4 \vee L_5))$$

$$\equiv ((L_1 \vee L_3) \wedge (L_2 \vee L_3)) \wedge [(L_1 \vee (L_4 \vee L_5)) \wedge (L_2 \vee (L_4 \vee L_5))]$$

$$\equiv (L_1 \vee L_3) \wedge (L_2 \vee L_3) \wedge (L_1 \vee L_4 \vee L_5) \wedge (L_2 \vee L_4 \vee L_5)$$

which is in CNF.

model: $\left\{ \begin{array}{l} x_1 + x_3 \geq 1 \\ x_2 + x_3 \geq 1 \\ x_1 + x_4 + x_5 \geq 1 \\ x_2 + x_4 + x_5 \geq 1 \end{array} \right\}$

## 2. Modeling with 0-1 and continuous variables

Let $y \in \{0,1\}$, $\bar{x} \in \mathbb{R}^n$

Statement : $y=1 \Rightarrow A\bar{x} \leq \bar{b}$

Assume $\exists\, \bar{u} \geq \bar{0}$ : $A\bar{x} \leq \bar{b} + \bar{u}$ is always true.

Then $\boxed{A\bar{x} \leq \bar{b} + \bar{u}(1-y)}$ is the model.

## 3. Modeling arbitrary disjunctions

$\bar{x} \in \mathbb{R}^n$

$$(A_1\bar{x} \leq \bar{b}^1) \vee (A_2\bar{x} \leq \bar{b}^2) \vee \cdots \vee (A_k\bar{x} \leq \bar{b}^k) \quad (*)$$

Assume $\{\bar{x} \mid A_i\bar{x} \leq \bar{b}^i\} \neq \emptyset$. if one system is $\emptyset$, then we could remove it from $(*)$

In words, $(*)$ says "$\bar{x}$ satisfies one of the $k$ systems"

Note that some of the statements using literals $L_i$ would fit this framework. At the same time, this is a much more general statement. We'll consider two approaches to model this statement. The first one looks quite similar to the previous case of $y=1 \Rightarrow A\bar{x} \leq \bar{b}$.

## big-M representation

**Assumption 1** $\exists\, \bar{u}^i \geq \bar{0}$ such that $\forall \bar{x}$ that satisfy $A_j \bar{x} \leq \bar{b}^j$ for some $j$, $A_i \bar{x} \leq \bar{b}^i + \bar{u}^i$ holds $\forall i$.

Let $y_i \in \{0,1\}$, $i = 1, \ldots, k$. → models whether the $i^{th}$ disjunction holds

$$\boxed{\begin{aligned} & A_i \bar{x} \leq \bar{b}^i + \bar{u}^i (1 - y_i), \quad i = 1, \ldots, k \\ & y_1 + y_2 + \cdots + y_k \geq 1 \\ & y_i \in \{0,1\}, \quad i = 1, \ldots, k. \end{aligned}}$$ —— (x-big M)

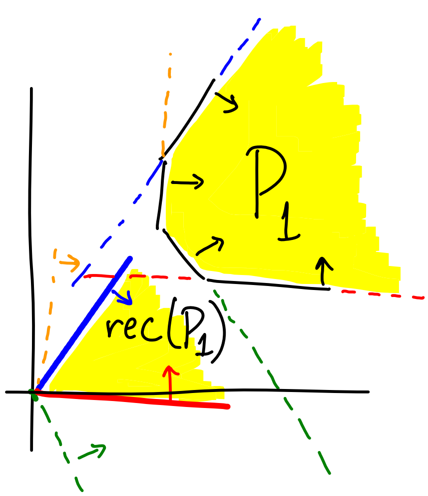
## Sharp formulation

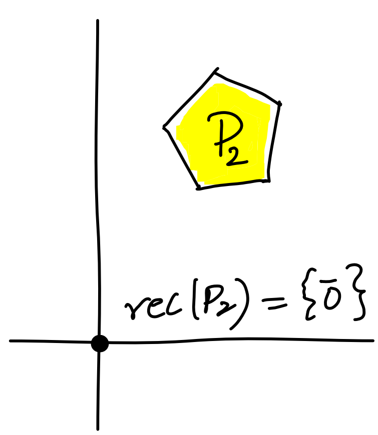
**Assumption 2** $\exists\, C$ such that

$C = \{\bar{x} \mid A_i \bar{x} \leq \bar{0}\}$, $i = 1, \ldots, k$ is independent of $i$.

**Def** The **recession cone** of polyhedron $P = \{\bar{x} \mid A\bar{x} \leq \bar{b}\}$ is

$$\mathbf{rec}(P) = \{\bar{x} \mid A\bar{x} \leq \bar{0}\}.$$




If $P$ is a polytope, i.e., a closed polyhedron, then $\mathrm{rec}(P) = \{\bar{0}\}$, the origin.

$$\boxed{\begin{aligned} A_1 \bar{x}^1 &\le \bar{b}^1 y_1 \\ &\vdots \\ A_k \bar{x}^k &\le \bar{b}^k y_k \\ \bar{x}^1 + \bar{x}^2 + \cdots + \bar{x}^k &= \bar{x} \\ y_1 + y_2 + \cdots + y_k &= 1 \\ y_i &\in \{0,1\} \end{aligned}} \quad (*\text{-sharp})$$

sharp, as exactly one (out of k) options is forced to hold

We now prove the correctness of $(*\text{-sharp})$.

**Theorem 1** $\bar{x}$ satisfies $(*)$ $\iff \exists (\bar{x}^1, \ldots, \bar{x}^k, y_1, \ldots, y_k)$ such that $(\bar{x}, \bar{x}^1, \ldots, \bar{x}^k, y_1, \ldots, y_k)$ satisfies $(*\text{-sharp})$.

**Proof** $(\Rightarrow)$ $\bar{x}$ satisfies $(*)$.

WLOG, let $A_1 \bar{x} \le \bar{b}^1$. We can choose

$$\left.\begin{aligned} &y_1 = 1,\ y_2 = \cdots = y_k = 0 \\ &\bar{x}^1 = \bar{x},\ \bar{x}^2 = \cdots = \bar{x}^k = \bar{0} \end{aligned}\right\} \text{ satisfies } (*\text{-sharp}).$$

$(\Leftarrow)$: in the next lecture...

# MATH 567: Lecture 5 (01/23/2025)

Today: * representing sets
* representing functions

## Proof of Theorem 1 (continued..)

Recall...

**Theorem 1** $\bar{x}$ satisfies (*) $\Longleftrightarrow$ $\exists\,(\bar{x}^1, \dots, \bar{x}^k, y_1, \dots, y_k)$ such that $(\bar{x}, \bar{x}^1, \dots, \bar{x}^k, y_1, \dots, y_k)$ satisfies (*-sharp).

$$\boxed{\begin{aligned} A_1 \bar{x}^1 &\le \bar{b}^1 y_1 \\ &\vdots \\ A_k \bar{x}^k &\le \bar{b}^k y_k \\ \bar{x}^1 + \bar{x}^2 + \cdots + \bar{x}^k &= \bar{x} \\ y_1 + y_2 + \cdots + y_k &= 1 \\ y_i &\in \{0,1\} \end{aligned}} \quad \text{(*-sharp)}$$

Proof ($\Rightarrow$): seen in the last lecture...

($\Leftarrow$) WLOG, let $y_1 = 1$, $y_i = 0$, $i = 2, \dots, k$ in $(\bar{x}, \bar{x}^1, \dots, \bar{x}^k, y_1, \dots, y_k)$ that satisfies (*-sharp).

$$\Rightarrow \quad \begin{aligned} A_1 \bar{x}^1 &\le \bar{b}^1 \\ A_2 \bar{x}^2 &\le \bar{0} \\ &\vdots \\ A_k \bar{x}^k &\le \bar{0} \end{aligned} \quad \text{and } \bar{x}^1 + \cdots + \bar{x}^k = \bar{x}$$

from Assumption 2

$$\Rightarrow \quad A_1 \bar{x} = A_1(\bar{x}^1 + \cdots + \bar{x}^k) = A_1 \bar{x}^1 + A_1 \bar{x}^2 + \cdots + A_1 \bar{x}^k \le \bar{b}^1 + \overbrace{\bar{0} + \cdots + \bar{0}}$$

$\Rightarrow A_1 \bar{x} \le \bar{b}^1$, i.e., $\bar{x}$ satisfies (*). $\square$

# Representing Sets in general

Q. In general, what all sets could we represent using 0-1 and/or general integer (GI) variables?

We need a few new definitions to address this question. In particular, we will formally define a formulation — so far, we have been studying them informally as MIP models.

**Def** A set $S$ is **bounded MIP-representable** (b-MIP-r) if $\exists$ matrices $A, B, C, D$ and a vector $\bar{f}$ such that

$$S = \{ (\bar{x}, \bar{y}) \in \mathbb{Z}^n \times \mathbb{R}^m \mid \exists (\bar{u}, \bar{v}) \in \mathbb{Z}^p \times \mathbb{R}^q \text{ such that } A\bar{x} + B\bar{y} + C\bar{u} + D\bar{v} \leq \bar{f} \}$$ and

$A\bar{x} + B\bar{y} + C\bar{u} + D\bar{v} \leq \bar{f}$ implies lower and upper bounds on $\bar{x}$ and $\bar{u}$ (the general integer (GI) variables). hence the "bounded" in b-MIP-r

The set $P = \{ (\bar{x}, \bar{y}, \bar{u}, \bar{v}) \in \mathbb{R}^{n+m+p+q} \mid A\bar{x} + B\bar{y} + C\bar{u} + D\bar{v} \leq \bar{f} \}$ is called a **formulation** of $S$.

Note that all variables are continuous in this formal definition of a formulation.

**Def** $P = \{ \bar{x} \in \mathbb{R}^n \mid A\bar{x} \leq \bar{b} \}$ is a **polyhedron**, where $A \in \mathbb{R}^{m \times n}$, $\bar{b} \in \mathbb{R}^m$, $m, n$ are finite. polyhedra are convex sets.

A bounded polyhedron is a **polytope**.

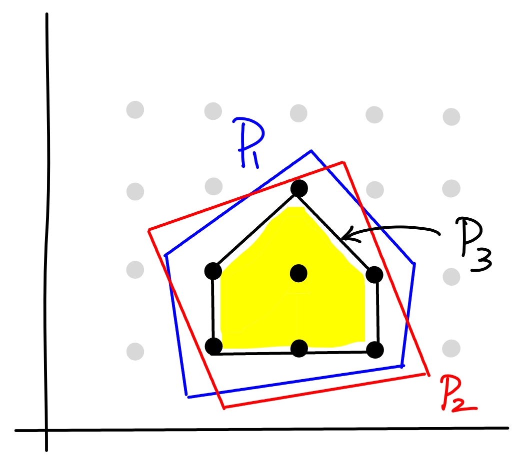
## Examples of formulations

1. $S = \{\bar{x} \in \{0,1\}^n \mid (x_1 = 1) \vee \cdots \vee (x_n = 1)\}$ has a formulation

$$P = \{\bar{x} \in \mathbb{R}^n \mid 0 \le x_i \le 1 \ \forall i, \ x_1 + \cdots + x_n \ge 1\}.$$

2. $S = \{7$ lattice points shown as $\bullet\}$.

$P_1, P_2, P_3$ are all formulations for $S$.

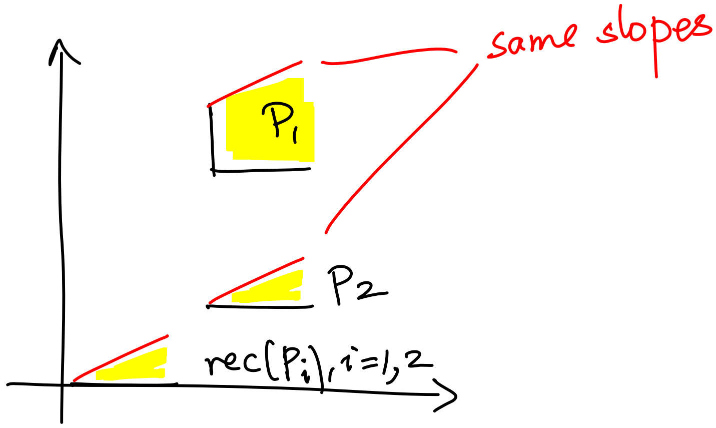



But $P_3$ is "better" than $P_1$ and $P_2$. Note that $P_3$ is the convex hull of the lattice points that is $S$. If we want to maximize a linear function over $S$, we could do the same over $P_3$ instead. The same claim cannot be made for $P_1$ or $P_2$. We will talk later about how to compare formulations.

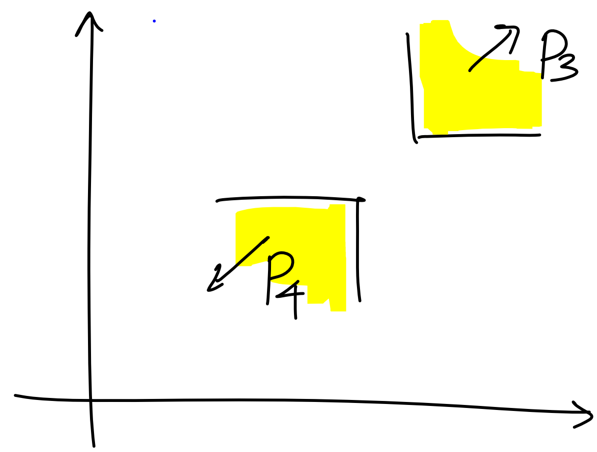
What kinds of sets $S$ are b-MIP-r? Could we characterize them?

**Theorem 2 (Jeroslow, Lowe):** $S$ is b-MIP-r iff $S = P_1 \cup \cdots \cup P_k$ for finite $k$, where $P_i$ are polyhedra having the same recession cone i.e., $\text{rec}(P_i)$ is independent of $i$, $i = 1, \ldots, k$.

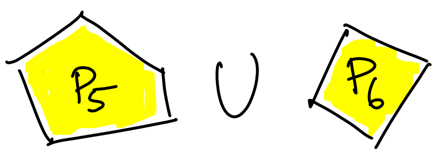
Here are some examples.



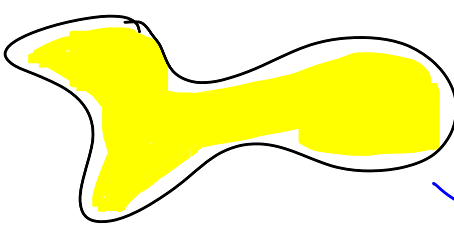

$P_1 \cup P_2$ is b-MIP-r.




$P_3 \cup P_4$ is not okay as an MIP.
(as $\text{rec}(P_3) \neq \text{rec}(P_4)$)

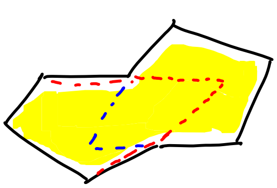
$S =$ 


is okay as an MIP.
$(\text{rec}(P_5) = \text{rec}(P_6) = \{\bar{0}\})$

$S =$ 

is not okay as MIP, as it is not a polyhedron to start with.

$S$ being non convex is not crucial here — it could've been an ellipse, and the conclusion is the same.

$S =$ 

o.k. as MIP! → union of two polytopes.

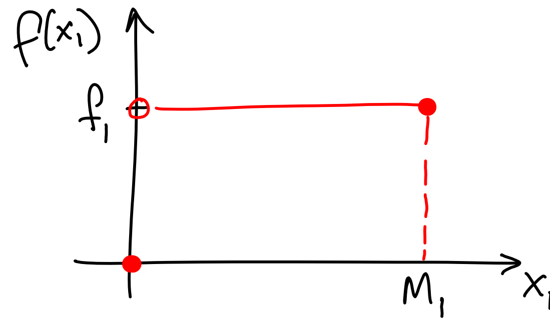
Can we use similar techniques to model functions, instead of sets?

## Representing Sets v/s Functions

We already saw the case of fixed charge:

$$\left\{\begin{array}{ll} \min & f(x_1) \\ \text{s.t.} & 0 \leq x_1 \leq M_1 \\ & A\bar{x} \leq \bar{b} \end{array}\right\}$$

and we wrote an MIP for the same.




It turns out we could use similar ideas to model some classes of functions appearing in certain optimization problems. Naturally, if $f(x_1)$ is nonlinear, e.g., $x_1^3$ or $\sqrt{x_1}$, we will not get an integer linear program! We need some definitions first.

**Def** Given $f: \mathbb{R}^n \to \mathbb{R}$, we define the **graph**, **epigraph**, and **hypograph** of $f$ as follows.

$$\text{graph}(f) = \{(z,\bar{x}) \in \mathbb{R}^{n+1} \mid z = f(\bar{x})\},$$

$$\text{epi}(f) = \{(z,\bar{x}) \in \mathbb{R}^{n+1} \mid z \geq f(\bar{x})\}, \text{ and}$$

$$\text{hypo}(f) = \{(z,\bar{x}) \in \mathbb{R}^{n+1} \mid z \leq f(\bar{x})\}.$$

Notice that graph$(f)$, epi$(f)$, and hypo$(f)$ are sets, and we could consider when each of them is b-MIP-r — instead of talking about representability of $f(\cdot)$ itself.

Suppose we have $\left\{\begin{array}{l} \min\ f(\bar{x}) \\ \text{s.t.}\ \ A\bar{x} \le \bar{b} \end{array}\right\}$ where epi(f) is b-MIP-r.

Then we can write $\left\{\begin{array}{l} \min\ z \\ \text{s.t.}\ \ A\bar{x} \le \bar{b} \\ \quad (z, \bar{x}) \in \text{epi}(f) \end{array}\right\}$ as the MIP representation.

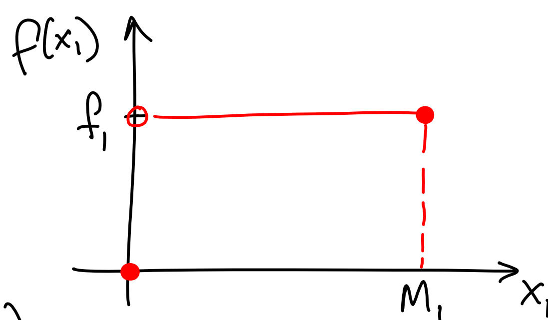
Since epi(f) is b-MIP-r, we can write down an MIP representation of epi(f), which completes the MIP model above.

Q. Why not require graph(f) being b-MIP-r?

**Theorem 3** graph(f) is b-MIP-r iff both epi(f) and hypo(f) are b-MIP-r.

## Example

The fixed charge function.

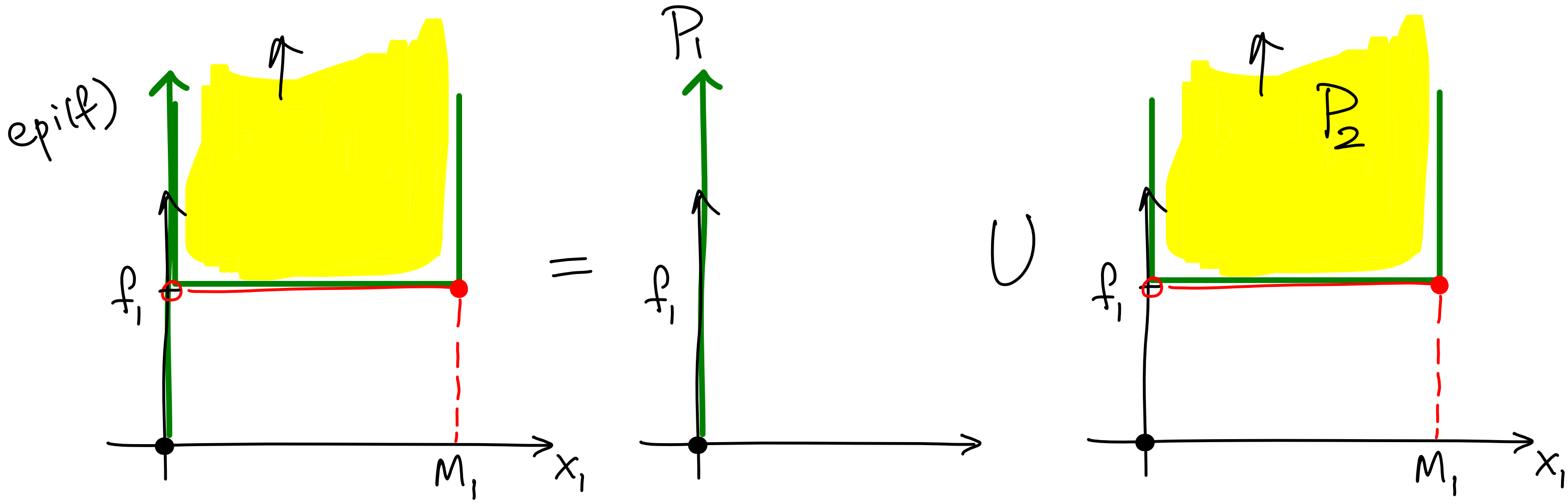



graph(f), i.e., $f(x_1)$ as drawn, is not b-MIP-r.

grap(f) is the union of origin (•) and and the half-open line segment (o———•).

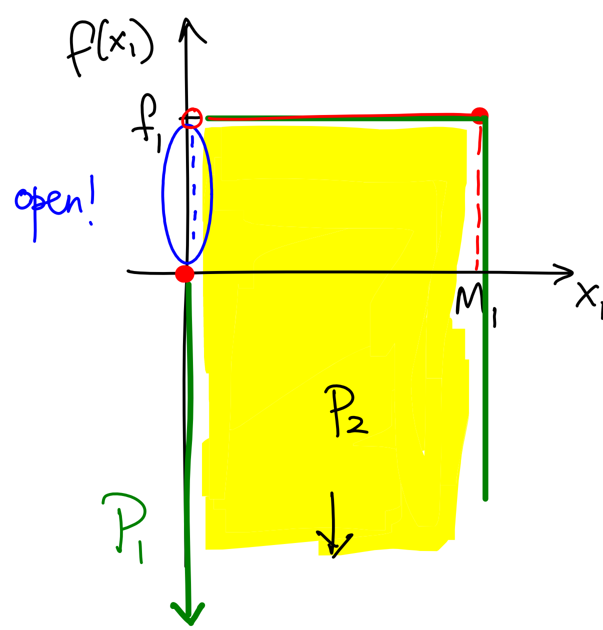
This second piece is not a polyhedron to start with.

But epi(f) turns out to be b-MIP-r here, and hypo(f) is not b-MIP-r at the same time.




Here $\operatorname{rec}(P_1) = \operatorname{rec}(P_2) = P_1$. Hence epi(f) is b-MIP-r.

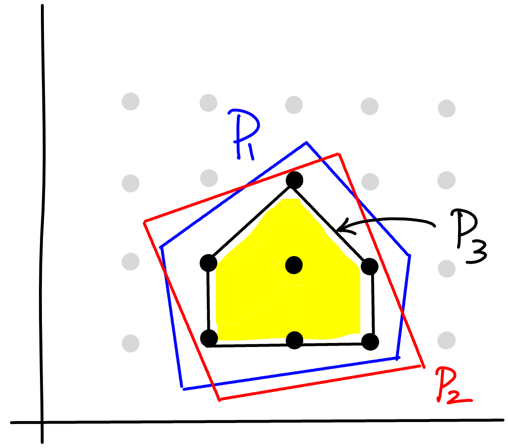
Here, hypo(f) is not b-MIP-r, as it is the union of $P_1$ and $P_2$, where $P_2$ is not a polyhedron (same reason as that for graph(f)).




What if $f_1 < 0$ here? Then hypo(f) will be b-MIP-r and epi(f) will not be b-MIP-r (the roles are reversed).

# Comparing Formulations

Q. What is a good/bad formulation?




We could say here $P_3$ is better than both $P_1$ and $P_2$, but cannot compare $P_1$ to $P_2$. More generally, we want to compare formulations with different sets (and hence different numbers) of extra variables. To this end, we need to introduce some basic results.

## Farkas' Lemma

pronounced "Farkash"; feasibility of alternative systems

① $\exists \bar{x} : A\bar{x} \le \bar{b}$ then

$A\bar{x} \le \bar{b}$ implies $\bar{a}^T\bar{x} \le \beta \iff$

$\exists \bar{u} \ge \bar{0}$ such that $\bar{u}^T A = \bar{a}^T$, $\bar{u}^T\bar{b} \le \beta$

multipliers using which we could derive $\bar{a}^T\bar{x} \le \beta$ from $A\bar{x} \le \bar{b}$

② $\exists \bar{x} : A\bar{x} \le \bar{b} \iff \nexists \bar{u} \ge \bar{0}$, $\bar{u}^T A = \bar{0}^T$, $\bar{u}^T\bar{b} < 0$.

(cannot derive $\bar{0}^T\bar{x} \le -1$ from $A\bar{x} \le \bar{b}$)

# MATH 567: Lecture 6 (01/28/2025)

Today: * comparing formulations
* sharp/ideal formulation

Recall Farkas' lemma. We present one more version now.

③ $\exists \bar{x} : A\bar{x} = \bar{b} \iff \not\exists \bar{u} \neq \bar{0} : \bar{u}^T A = \bar{0}^T, \; \bar{u}^T \bar{b} = -1$ can be any nonzero #

$A\bar{x} = \bar{b}$ : $[A|\bar{b}] \xrightarrow{\text{EROs}} [\quad]$ echelon form

If the echelon form has as row of the form $[0\,0 \cdots 0 | \blacksquare]$, $\blacksquare \neq 0$, the system $A\bar{x} = \bar{b}$ is inconsistent.

Naturally, we can prove the other versions if we assume one version of Farkas' lemma (i.e., they are equivalent).

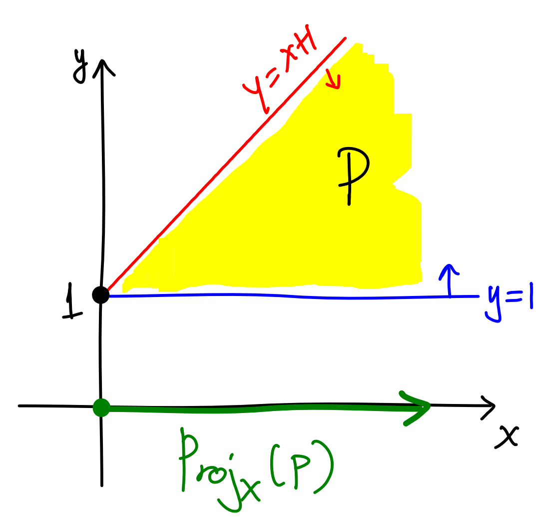
**Def** If $P = \{(\bar{x}, \bar{y}) \,|\, A\bar{x} + B\bar{y} \leq \bar{b}\}$, then the **projection of P** onto the space of $\bar{x}$ variables is

$$\text{Proj}_{\bar{x}}(P) = \{\bar{x} \,|\, \exists \bar{y} : (\bar{x}, \bar{y}) \in P\}.$$

⇓ "push down" P onto the x-axis

e.g., $P = \{(x,y) \,|\, y \geq 1, \, y \leq x+1\}$

$$\text{Proj}_x(P) = \{x \,|\, x \geq 0\}.$$



**Theorem 4** $\mathrm{Proj}_{\bar{x}}(P) = \{\bar{x} \mid \underbrace{\bar{v}^T A\bar{x} \le \bar{v}^T\bar{b} \ \forall \bar{v} \ge \bar{0},\ \bar{v}^T B = \bar{0}^T\}}_{(RHS)}$.

In words, all nonnegative linear combinations of $A\bar{x} + B\bar{y} \le \bar{b}$ that eliminate the "unwanted" $\bar{y}$ variables.

**Proof** '$\subseteq$': $\bar{x} \in \mathrm{Proj}_{\bar{x}}(P) \Rightarrow \exists \bar{y} \mid A\bar{x} + B\bar{y} \le \bar{b}$

$\Rightarrow \bar{v}^T A\bar{x} \le \bar{v}^T\bar{b}$ holds $\forall \bar{v} \ge 0$, $\bar{v}^T B = \bar{0}^T$.

'$\supseteq$': Show that if $\bar{x} \notin \mathrm{Proj}_{\bar{x}}(P)$, then $\bar{x} \notin (RHS)$.

Can use Farkas' lemma!

$\nexists \bar{y}: B\bar{y} \le \bar{b} - A\bar{x}$, i.e., the system $B\bar{y} \le \bar{b} - A\bar{x}$ has no solutions (in $\bar{y}$).

$\Rightarrow \exists \bar{v} \ge \bar{0}$, $\bar{v}^T B = \bar{0}^T$ and $\bar{v}^T(\bar{b} - A\bar{x}) < 0$.

$\Rightarrow \exists \bar{v} \ge \bar{0}$, $\bar{v}^T B = \bar{0}^T$ for which $\bar{v}^T A\bar{x} > \bar{v}^T\bar{b}$.

$\Rightarrow \bar{x} \notin (RHS)$. $\square$

Back to 2D example:

$$\begin{matrix} y \ge 1 \\ y \le x+1 \end{matrix} \Rightarrow \left.\begin{matrix} -y \le -1 \\ y \le x+1 \end{matrix}\right\} \xRightarrow{ADD} x \ge 0,$$ which is $\mathrm{Proj}_x(P)$.

Equivalently, $\left\{ \begin{array}{r} -y \le -1 \\ -x+y \le 1 \end{array} \right\} \equiv \underbrace{\begin{bmatrix} 0 \\ -1 \end{bmatrix}}_{A} x + \underbrace{\begin{bmatrix} -1 \\ 1 \end{bmatrix}}_{B} y \le \underbrace{\begin{bmatrix} -1 \\ 1 \end{bmatrix}}_{\bar{b}}$.

What $\bar{v} \geq \bar{0}$ with $\bar{v}^T B = 0$ can we take to eliminate $y$?

$\bar{v} = \begin{bmatrix} \lambda \\ \lambda \end{bmatrix}$, $\lambda \geq 0$ works! Or, $\bar{v} = \lambda \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $\lambda \geq 0$ are all the multipliers!

Could we generalize this result illustrated in the example, i.e., could we describe $\mathrm{Proj}_{\bar{x}}(P)$ using only a finite # $\bar{v}^i$'s?

**Theorem 5** If $\{\bar{v} \geq \bar{0},\ \bar{v}^T B = \bar{0}^T\} = \underbrace{\textbf{cone}\{\bar{v}^1, \dots, \bar{v}^k\}}_{\text{the projection cone is finitely generated}}$

$\stackrel{\text{def}}{\equiv} \left\{ \sum_{i=1}^{k} \lambda_i \bar{v}^i \,\middle|\, \lambda_i \geq 0 \right\}$, then

$$\mathrm{Proj}_{\bar{x}}(P) = \left\{ \bar{x} \,\middle|\, (\bar{v}^i)^T A \bar{x} \le (\bar{v}^i)^T \bar{b},\ i = 1, \dots, k \right\}.$$

**Def** The set $\{ \bar{v} \mid \bar{v} \geq \bar{0},\ \bar{v}^T B = \bar{0}^T \}$ is called the **projection cone** of $\mathrm{Proj}_{\bar{x}}(P)$.

Example (continued): The projection cone of $\mathrm{Proj}_x(P)$ is

$$\left\{ \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \,\middle|\, \begin{array}{l} v_1, v_2 \geq 0 \\ -v_1 + v_2 = 0 \end{array} \right\} = \left\{ \lambda \begin{bmatrix} 1 \\ 1 \end{bmatrix} \,\middle|\, \lambda \geq 0 \right\}.$$

# Definition of Comparison

Let $S \subseteq \mathbb{Z}^n \times \mathbb{R}^m$ have two formulations

$$P_1 = \{(\bar{x}, \bar{y}, \bar{u}^1, \bar{v}^1) \in \mathbb{R}^{n+m+p_1+q_1} \mid A_1\bar{x} + B_1\bar{y} + C_1\bar{u}^1 + D_1\bar{v}^1 \leq \bar{b}^1\}$$

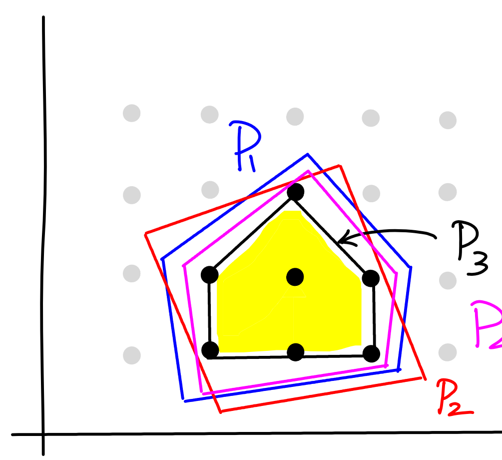
and

$$P_2 = \{(\bar{x}, \bar{y}, \bar{u}^2, \bar{v}^2) \in \mathbb{R}^{n+m+p_2+q_2} \mid A_2\bar{x} + B_2\bar{y} + C_2\bar{u}^2 + D_2\bar{v}^2 \leq \bar{b}^2\}$$

where $p_1 \neq p_2$, $q_1 \neq q_2$ and $p_1+q_1 \neq p_2+q_2$.

Then $P_1$ is a **better (stronger, tighter)** formulation than $P_2$ if $Proj_{(\bar{x},\bar{y})}(P_1) \subset Proj_{(\bar{x},\bar{y})}(P_2)$.

$P_j$, $j=1,2,3,4$ are formulations of $S$.




Here, $P_3 \subset P_j$, $j=1,2,4$.

So, $P_3$ is stronger than $P_j$.

Similarly, $P_4 \subset P_1$, so is stronger than $P_1$.

## Example

$$S = \{\bar{x} \mid \bar{x} \in \{0,1\}^n, (x_1=1) \Rightarrow (x_2=1) \wedge \cdots \wedge (x_n=1)\}.$$

$$P_1 = \{\bar{x} \mid x_1 \le x_2, \ldots, x_1 \le x_n, \quad 0 \le x_i \le 1, \quad i=1,\ldots,n\} \text{ and}$$

$$P_2 = \{\bar{x} \mid (n-1)x_1 \le x_2 + \cdots + x_n, \quad 0 \le x_i \le 1, \quad i=1,\ldots,n\}$$

are formulations for $S$. ($P_i \cap \mathbb{Z}^n$ gives $S$ for $i=1,2$).

$P_1$ is the **disaggregated** formulation, while $P_2$ is an **aggregated** formulation.

**Claim** $P_1 \subset P_2$.

$P_1 \subseteq P_2$ is trivial – just add up $x_1 \le x_i, i=2,\ldots,n$, to get $(n-1)x_1 \le x_1 + \cdots + x_2$.

To show $P_1 \subset P_2$, identify one point in $P_2 / P_1$.

$\left(\frac{1}{n-1}, 1, 0, \ldots, 0\right) \in P_2 / P_1$. For instance, $x_3 \ge x_1$ is violated here.

In fact, $P_1$ is the strongest formulation for $S$ here!

**Def** Given $S \in \mathbb{Z}^n \times \mathbb{R}^m$, $P \subseteq \mathbb{R}^n \times \mathbb{R}^m$ is a **sharp** or **ideal formulation** for $S$ if

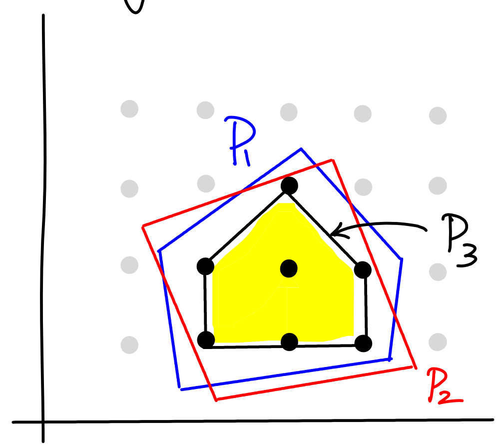
(1) $\forall \begin{bmatrix} \bar{c} \\ \bar{d} \end{bmatrix} \in \mathbb{R}^{n+m}$ such that $\max \left\{ [\bar{c}^T \bar{d}^T] \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} \,\middle|\, \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} \in P \right\}$

is finite, the optimum is obtained for some element of $S$.

(2) An extended formulation of $S$ (using extra variables) is sharp if its projection to $(\bar{x}, \bar{y})$-space is sharp in the sense of (1) above.

Intuitively, all corner points of $P$ are integral.

$P_3$ is the sharp formulation of $S$ here:




More generally, $P$ is the convex hull of $S$.

**Def** For $X \subseteq \mathbb{R}^n$, the **convex hull** is defined as

$$\text{conv}(X) = \left\{ \bar{x} \in \mathbb{R}^n \,\middle|\, \bar{x} = \sum_{i=1}^{k} \lambda_i \bar{x}^i,\ \lambda_i \geq 0,\ \sum_{i=1}^{k} \lambda_i = 1, \text{ for all finite subsets } \{\bar{x}^1, \dots, \bar{x}^k\} \text{ of } X \right\}.$$

To show a formulation P is sharp for a set S, we can show every corner point of P is integral, i.e., all their entries are integers.

In 2D, any two non-parallel lines representing equations from P **could** intersect at a corner point, assuming it is feasible.

In general, in $\mathbb{R}^n$, we get a corner point from $n$ linearly independent (LI) equations that define P, assuming their intersection is feasible.

We saw that $\left(\frac{1}{n-1}, 1, 0, 0 \ldots, 0\right) \in P_2$ (aggregated formulation).
In fact, this point is a corner point of $P_2$, defined by the $n$ LI constraints.

→ more on this and other details in the next lecture...

# MATH 567: Lecture 7 (01/30/2025)

Today: * sharp formulations
* TSP Formulations

**Example** $\left[ (x_1 = 1) \Rightarrow \bigwedge_{j=2}^{n} (x_j = 1),\ x_i \in \{0,1\} \forall i \right]$, $P_1, P_2$, (continued...)

We saw that $\left(\frac{1}{n-1}, 1, 0, 0 \ldots, 0\right) \in P_2$ (aggregated formulation). In fact, this point is a corner point of $P_2$, defined by the $n$ LI constraints

$$(n-1)x_1 \leq x_2 + \cdots + x_n \quad \text{--- (1)}$$
$$x_2 \leq 1 \quad \text{--- (2)}$$
$$x_j \geq 0, j = 3, \ldots, n \quad \text{--- (n-2)}$$

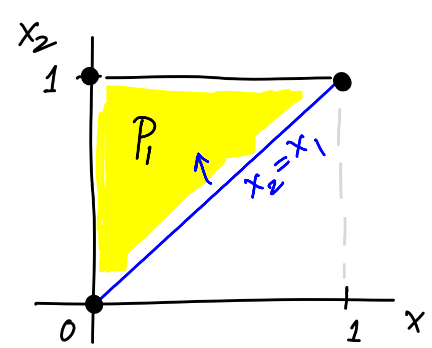
satisfied as equations. Hence $P_2$ is not sharp for $S$.

To show $P_1$ (disaggregated formulation) is sharp for $S$, we show each corner point of $P_1$ is integral. The inequalities defining $P_1$ can be broken down into 3 groups:

$$x_i \geq x_1, \quad i = 2, \ldots, n \quad \text{--- (1)}$$
$$x_i \geq 0, \quad i = 1, \ldots, n \quad \text{--- (2)}$$
$$x_i \leq 1, \quad i = 1, \ldots, n \quad \text{--- (3)}$$

First, check intuition in 2D:

$$P_1 = \{ \bar{x} \in \mathbb{R}^2 \mid x_2 \geq x_1,\ 0 \leq x_i \leq 1,\ i = 1,2 \}.$$




Indeed, all three corner points are integral!

In general, we consider a few cases:

(i) All $(n-1)$ inequalities from (1), and one from (2) or (3):

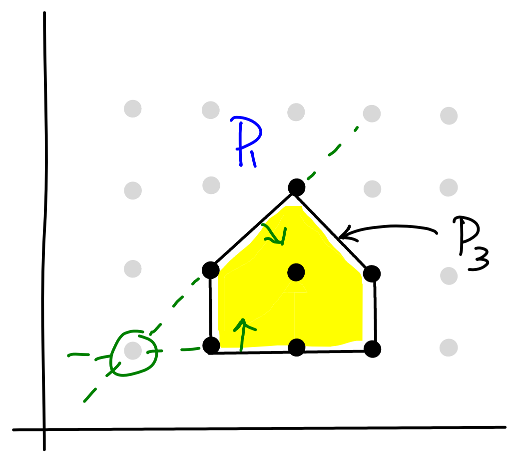
$$\Rightarrow \quad x_i = 0 \ \forall i \quad \text{or} \quad x_i = 1 \ \forall i.$$

(ii) All $n$ inequalities from (2) or (3): $\Rightarrow$ trivial.

(iii) $2 \leq j < n$ inequalities from (1) and $n-j$ inequalities from (2) or (3) $\Rightarrow$ Can show $x_i \in \{0,1\} \ \forall i$.

Recall that we need to consider equality versions of the constraints and their intersections to enumerate all potential corner points.

For instance, the two LI lines corresponding to two constraints indicated by green dashed lines here meet at a point that is not feasible, and hence is not a corner point of $P_3$.




If we can provide a sharp formulation with a "small", i.e., a polynomial number, of inequalities, then we can solve any linear optimization problem over $S$ "easily", i.e., in polynomial time.

But for many problems, e.g., the traveling salesman problem (TSP), the sharp formulation has exponentially many inequalities.

We now consider several examples of sharp formulation.

# More Examples of Sharp Formulations

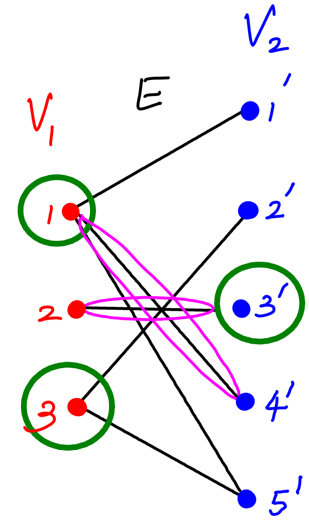
1. Given a bipartite graph $G = (V_1 \cup V_2, E)$, let $S_M \subseteq \mathbb{Z}^{|E|}$ be the collection of incidence vectors of all matchings. M for matching

Then $P_m = \left\{ \bar{x} \in \mathbb{R}^{|E|} \;\middle|\; \bar{x} \geq \bar{0}, \right.$

$$\sum_{e \ni i} x_e \leq 1 \;\; \forall i \in V_1,$$

$$\left. \sum_{e \ni j} x_e \leq 1 \;\; \forall j \in V_2 \right\}$$

is a sharp formulation for $S_m$.




$\{(1,4'),(2,3')\}$ is a matching, and $\{1,3,3'\}$ is a node cover.

2. Given $G = (V_1 \cup V_2, E)$ as above, let $S_N \subseteq \mathbb{Z}^{|V|}$ be the collection of incidence vectors of all node covers, i.e., subsets of nodes that cover all edges. Then

$V = V_1 \cup V_2$

$$P_N = \left\{ \bar{x} \in \mathbb{R}^{|V|} \;\middle|\; 0 \leq \bar{x} \leq \bar{1}, \sum_{i \in e} x_i \geq 1 \;\forall e \in E \right\}$$

$x_i + x_j \geq 1 \;\; \forall (i,j) \in E$

is a sharp formulation for $S_N$.

The default problems ask to identify the maximum (cardinality) matching, and the minimum (cardinality) node cover.

3. Define $S_{ij}$ and $P_{ij}$ as follows, using $x_1, x_2, x_3 \in \{0,1\}$.

$$S_{ij} = \{\bar{x} \mid \bar{x} \in \{0,1\}^3, \ (x_i = 0) \vee (x_j = 0)\} \text{ and}$$

$$P_{ij} = \{\bar{x} \mid \bar{x} \in \mathbb{R}^3, \ 0 \leq \bar{x} \leq \bar{1}, \ x_i + x_j \leq 1\}, \text{ for } i,j = 1,2,3, \ i \neq j.$$

We can show $P_{ij}$ is a sharp formulation for $S_{ij}$.

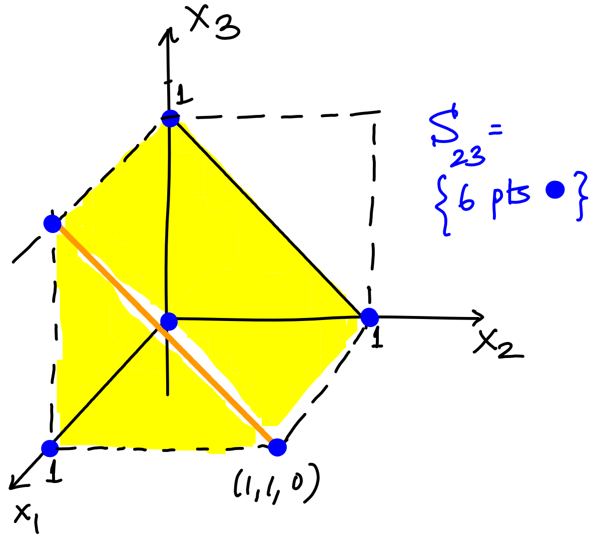
For instance,

$$S_{23} = \{(0,0,0), (1,0,0), (0,1,0), (1,1,0), (0,0,1), (1,0,1)\} \text{ and}$$

$$P_{23} = \left\{ \bar{x} \in \mathbb{R}^3 \;\middle|\; x_2 + x_3 \leq 1, \ \begin{array}{l} x_1 \geq 0, \ x_2 \geq 0, \ x_3 \geq 0 \\ x_1 \leq 1, \ \underbrace{x_2 \leq 1, \ x_3 \leq 1}_{\text{could remove!}} \end{array} \right\}.$$

Could check all subsets of 3 LI inequalities $\left( \leq \binom{7}{3} = 35 \text{ choices} \right)$.

We could also just plot $P_{23}$!




In another formulation, we could drop $x_2 \leq 1$ and $x_3 \leq 1$, since we have $x_2 + x_3 \leq 1$, which implies $x_2 \leq 1$ and $x_3 \leq 1$ along with $x_2 \geq 0$ and $x_3 \geq 0$.

Now, let

$$S = S_{12} \cap S_{23} \cap S_{31} = \{ \bar{x} \in \{0,1\}^3 \mid (x_1=0) \vee (x_2=0) \wedge (x_2=0) \vee (x_3=0) \wedge (x_3=0) \vee (x_1=0) \}$$

and

$$P = P_{12} \cap P_{23} \cap P_{31} = \{ \bar{x} \in \mathbb{R}^3 \mid 0 \le \bar{x} \le \bar{1},\ \underline{x_1+x_2 \le 1,\ x_2+x_3 \le 1,\ x_3+x_1 \le 1} \}.$$

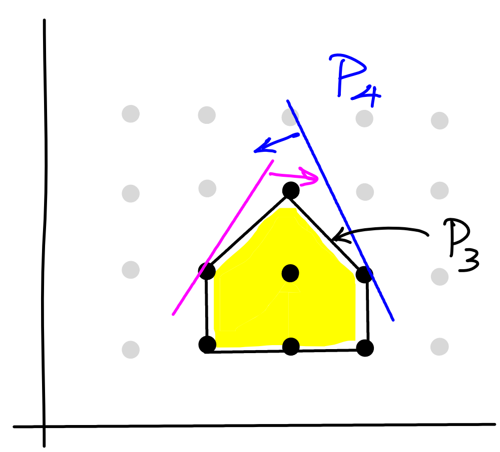
Is $P$ a sharp formulation for $S$? No!

For instance, $\max \{ x_1+x_2+x_3 \mid \bar{x} \in P \}$ has a unique optimal solution at $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}) \notin S$.

Notice that in $S$, no point can have two $x_j$'s set to 1, as $(x_i=0) \vee (x_j=0)$ is true for all three pairs. So we cannot get $x_1+x_2+x_3 = 2$. But $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}) \in P$, and indeed gives a higher value for $x_1+x_2+x_3$.

Also, $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ is a corner point of $P$: it is the point of intersection of $x_1+x_2=1$, $x_2+x_3=1$, $x_3+x_1=1$, which are LI.

There was question about whether the sharp formulation $P$ is unique. As a set, it captures the convex hull of $S$, and $\text{conv}(S)$ itself is unique. But there could be alternative descriptions of $P$, e.g., by adding redundant constraints, as shown here with the case of $P_4$, which adds two redundant constraints to $P_3$.



# Traveling Salesman Problem (TSP)

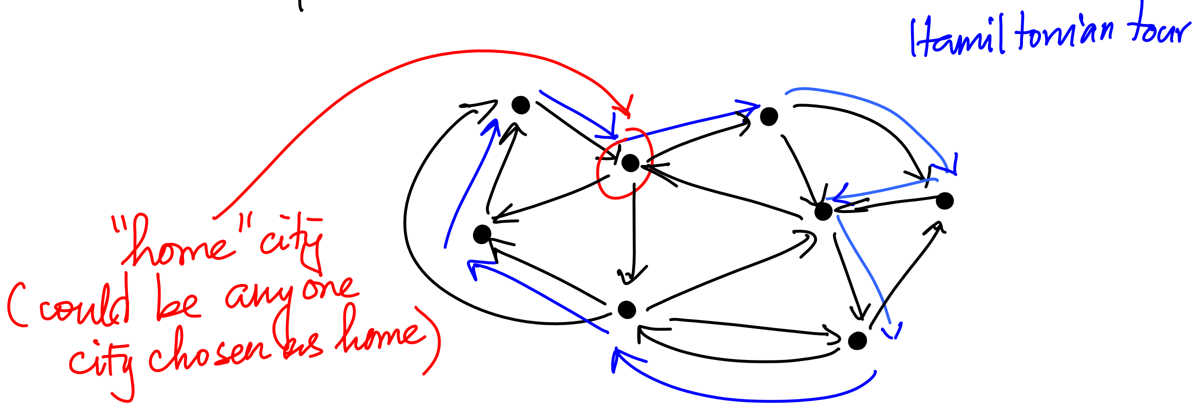
* $n$ cities
* $c_{ij}$ : cost (or distance) from city $i$ to city $j$ is defined on a directed graph $G = (V, E)$.
  ↳ directed

Goal: find a ~~shortest~~ smallest total costs Hamiltonian tour, i.e., a single directed cycle that contains all $n$ nodes (cities), and each node is visited exactly once.

TSP is perhaps the most widely studied combinatorial optimization problem. We will consider a few different formulations for the TSP.

Forget $c_{ij}$'s for now.

Goal: Formulations for $S \subseteq \mathbb{Z}^{|E|}$, the set of incidence vectors of all Hamiltonian tours




$$S = \{ \bar{x} \mid \bar{x} \text{ is the incidence vector of a Hamiltonian tour} \}.$$

$x_{ij} = 1$ if $(i,j) \in$ tour

$\bar{x} \in S \Rightarrow$

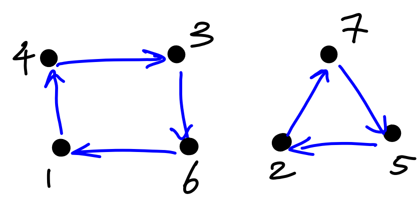
$$\left.\begin{array}{l} \sum_{j:(i,j)\in E} x_{ij} = 1 \quad \forall i \\ \sum_{j:(i,j)\in E} x_{ji} = 1 \quad \forall i \\ 0 \le x_{ij} \le 1, \quad \bar{x} \in \mathbb{Z}^{|E|} \end{array}\right\} \text{———— (1)}$$

remove to get formulation, i.e., the polytope

Assume $x_{ii} = 0 \ \forall i$.

But (1) is not enough, as it allows subtours.

Here is a collection of two subtours, which together satisfy (1):




We have to avoid subtours. We examine a few different ways to avoid them. One option involves adding some extra variables, and extra constraints. The other option involves adding extra constraints using the original variables ($x_{ij}$).

We have to avoid subtours!

<u>First approach</u> We add $u_i$, $i = 1, \ldots, n$, node variables.

$u_i \equiv$ position of node $i$ in tour.

any node could be the "home city"

We assume node 1 is the "home city". $u_1 = 1$.

$u_3 = 5 \Rightarrow$ Node 3 is the $5^{th}$ node in the tour, starting from node 1.

# MATH 567 : Lecture 8 (02/04/2025)

Today: TSP formulations and comparisons

**Recall** $u_i \equiv$ position of node $i$ in tour.

We want to impose

if $x_{ij} = 1$ then $u_j \geq u_i + 1$ for $i \neq 1, j \neq 1$.

We write

$$u_i - u_j + 1 \leq n(1 - x_{ij}), \; \forall i \neq 1, \forall j \neq 1 \qquad (2)$$

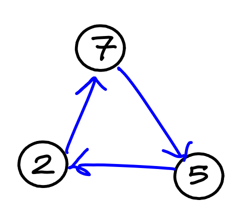
Let's check:

if $x_{ij} = 1$ (2) $\Rightarrow$ $u_i - u_j + 1 \leq 0 \Longrightarrow u_j \geq u_i + 1$. ✓

if $x_{ij} = 0$ (2) $\Rightarrow$ $u_i - u_j \leq n - 1$. ✓

Notice that $u_j$ need not represent the position of node $j$ in the tour exactly. But $u_j$ will be at least $u_i + 1$ when $x_{ij} = 1$. Thus, we could have $u_j = u_i + 5$, for instance. But even such values eliminate subtours, as they will not allow split (sub)tours as illustrated previously.

$$\{ u_7 \geq u_2 + 1, \; u_2 \geq u_5 + 1, \; u_5 \geq u_7 + 1 \}$$



Cannot hold together!

But if we add $2 \leq u_j \leq n, \; \forall j \neq 1$, we get $u_j$ representing the position of node $i$ exactly.

**Claim** $S = \{\bar{x} \in \mathbb{Z}^{|E|} \mid \exists \bar{u} : (\bar{x}, \bar{u}) \text{ satisfies (1) and (2)}\}$.

**Proof** '$\subseteq$': If $\bar{x}$ is a tour, take $u_i =$ position of node $i$ in $\bar{x}$.

If $x_{ij} = 1$ (2) $\Rightarrow$ $u_j \geq u_i + 1$. ✓

$x_{ij} = 0$ (2) $\Rightarrow$ $u_i - u_j + 1 \leq n$. ✓

want to show

'$\supseteq$': $\bar{x} \notin S \Rightarrow \bar{x}$ violates (1) or
$\bar{x}$ satisfies (1), but there is no $\bar{u}$ to satisfy (2).

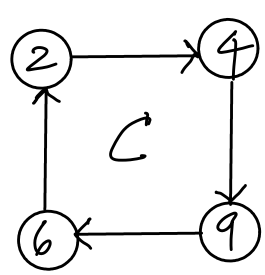
Case 1: $\bar{x}$ violates (1): trivial.

Case 2: $\bar{x}$ satisfies (1), but is not a tour.

Let $C$ be a subtour with $1 \notin C$. In more detail,

$$C = \Big\{ \overbrace{\{i_1, i_2, \ldots, i_k\}}^{\text{nodes}} \text{ along with edges } (i_r, i_{r+1}),\ r = 1, \ldots, k-1 \text{ and } (i_k, i_1), \text{ where } i_r \neq 1. \Big\}$$

e.g.,




Consider

$$u_i - u_j + 1 \leq n(1 - x_{ij}) \quad \text{——(2)}$$

for each $i, j \in C$.

Add (2) around $C \Rightarrow |C| \leq n\,(|C| - X(C))$ where

$X(C) = \sum_{(i,j) \in C} x_{ij}$. Hence $X(C) \leq \left(1 - \frac{1}{n}\right)|C|$.

But $x_{ij}$'s violate this inequality! there will be exactly $|C|$ $x_{ij}$'s set to 1!

$\square$

**Remark** If we use (1), and instead of (2), write

$$\left.\begin{aligned} &1 \le u_i \le n && \text{(a)} \\ &u_i - u_j + 1 \le n(1 - x_{ij}), \ \forall i, \ \forall j \ne 1 && \text{(b)} \\ &n - u_i \le (n-1)(1 - x_{i1}), \ \forall i \ne 1 && \text{(c)} \end{aligned}\right\} \quad (3),$$

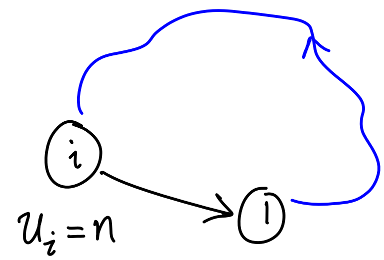
then (1) & (3) together give a valid formulation for $S$.

3 (b) forces $x_{ij} = 1 \Rightarrow u_j \ge u_i + 1, \ \forall i, \ \forall j \ne 1$.

3 (c) forces $X_{i1} = 1 \Rightarrow u_i \ge n \quad \forall i \ne 1$.

forces $u_i = n$, with 3(a)

$u_i$ for node $i$ in the are $(i, 1)$ coming into node 1 is forced to $n$, making $i$ the last node in the tour.




$$S = \left\{ \bar{x} \in \mathbb{Z}^{|E|} \;\middle|\; \exists \bar{u} : (\bar{x}, \bar{u}) \text{ satisfy (1) \& (3)} \right\}.$$

(1)+(2) and (1)+(3) are quite similar to each other in terms of strength, as well as in computation.

(1)+(2) is the **Miller-Tucker-Zemlin (MTZ)** formulation.

# Subtour Formulation

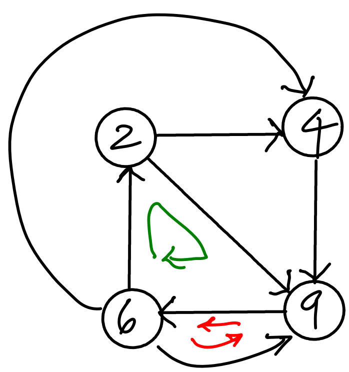
$$\boxed{\forall\, W \subseteq V,\ |W|>1,\ \sum_{\substack{i,j\in W\\(i,j)\in E}} x_{ij} \le |W|-1} \quad \text{———} (4)$$

more on this point in a bit...

(4) has exponentially many constraints in $|V|=n$.

(1)+(4) is a valid formulation for $S$, i.e.,

$$S = \{\bar{x} \in \mathbb{Z}^{|E|} \mid \bar{x} \text{ satisfies (1) and (4)}\}.$$



$W=\{2,4,6,9\}$

$$\underbrace{\sum_{\substack{i,j\in W\\(i,j)\in E}}}_{7\ x_{ij}\ \text{terms here}} x_{ij} \le 3$$

This constraint will avoid **all** possible subtours of length 4 in $G$ which use $\{2,4,6,9\}$, and not just the obvious one, i.e., 2-4-9-6-2.

At the same time, this constraint will allow subtours of length 2 or 3 in $W$, e.g., 6-9-6 or 2-9-6-2. We need the subtour constraints for $W'=\{6,9\}$ and $W''=\{2,6,9\}$ to eliminate them.

Now, let's consider the $W \underset{?}{\subsetneq} V$ question...

**Q.** Should we write the subtour constraint for $W = V$? Wouldn't that eliminate all possible Hamiltonian tours?

The answers are YES and YES, as it does not matter much when considering formulations for $S$. The default option is that we write the subtour constraints for all $W \subset V$, i.e., with $|W| \leq n-1$. In this case, we will indeed capture the Hamiltonian tours.

On the other hand, we could write the subtour constraint for $W = V$, in which case the Hamiltonian tours are avoided. But Hamiltonian paths are still permitted, and we could add the last missing arc in any Hamiltonian path to get the corresponding tour.

But once we involve the costs $c_{ij}$, we should ideally not write the subtour constraint for $W = V$. The last connecting arc (to complete the tour) could have a huge cost, affecting the minimality computations.

Also, notice that (4) is valid for $|W| = 1$, since we assume that there are no self loops, i.e., no arcs $(i,i)$. Equivalently, $x_{ii} = 0 \ \forall i$.

# Comparing MTZ and Subtour formulations

## First guess

If $C$ is a subtour (cycle) with $1 \notin C$, adding (2) around $C$ got us

$$X(C) \leq \left(1-\frac{1}{n}\right)|C| \qquad (\triangle)$$

If $n=100$, $X(C) \leq 0.99|C|$, which is not very effective.

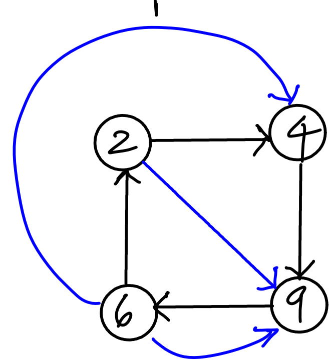
Notice that $X(C) \leq |C|$ holds trivially (and from (1)). So, as $n$ becomes larger and larger, the right-hand side value becomes closer and closer to $|C|$, while still remaining strictly smaller than $|C|$.

In the subtour formulation, using $W=C$, we get

$$\underbrace{\sum_{\substack{i,j \in W \\ (i,j)\in E}}}_{\text{more } x_{ij} \text{ terms than included in } X(C).} x_{ij} \leq \underbrace{|C|-1}_{\text{almost "1 better than } (\triangle)\text{"}.} \qquad (*)$$

Thus $(*)$ is stronger than $(\triangle)$ in both the left-hand and the right-hand sides. But we now make this comparison more formal.

not just the 4 arcs in $C$ (2-4-9-6-2)



We consider

$P_{MTZ} = \{ (\bar{x}, \bar{u}) \in \mathbb{R}^{|E|} \times \mathbb{R}^{|V|} \mid (\bar{x}, \bar{u}) \text{ satisfy (1) and (2)} \}$, and

$P_{subtour} = \{ \bar{x} \in \mathbb{R}^{|E|} \mid \bar{x} \text{ satisfies (1) and (4)} \}$.

To compare, we compute $\mathrm{Proj}_{\bar{x}}(P_{MTZ})$.

**Theorem 6** $\mathrm{Proj}_{\bar{x}}(P_{MTZ}) = \{ \bar{x} \mid \exists \bar{u} : (\bar{x}, \bar{u}) \text{ satisfies (1) \& (2)} \}$

$$= \{ \bar{x} \mid \bar{x} \text{ satisfies } (\triangle) \text{ for all cycles } C,\ 1 \notin C \}.$$

there could be exponentially many such cycles.

Indeed, $P_{MTZ}$ is described by a small (polynomial in $m, n$) number of constraints using the $n$ extra variables $u_i$. But an exponential number of constraints are needed to describe $\mathrm{Proj}_{\bar{x}}(P_{MTZ})$.

**Proof** $P_{MTZ}$ is given by (1) and

$$u_i - u_j + 1 \le n(1 - x_{ij}) \quad \forall i \ne 1, \forall j \ne 1. \quad \text{——— (2)}$$

$$\Rightarrow \quad u_i - u_j + n x_{ij} \le n - 1, \quad \forall i \ne 1, \forall j \ne 1.$$

**Recall:** Projection of $P = \{ (\bar{x}, \bar{y}) \mid A\bar{x} + B\bar{y} \le \bar{b} \}$ to $\bar{x}$:

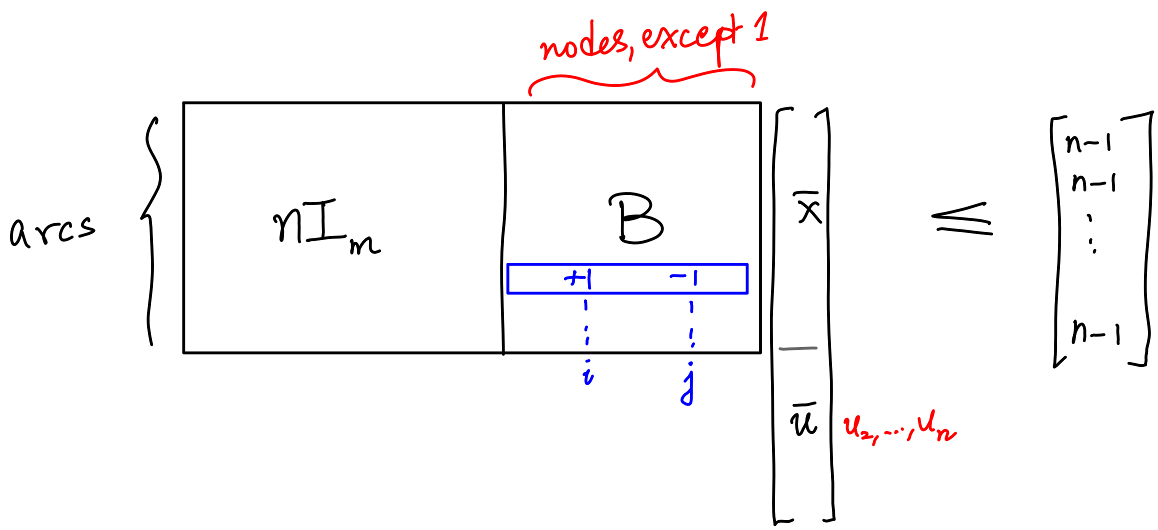
$$\mathrm{Proj}_{\bar{x}}(P) = \{ \bar{x} \mid \exists \bar{y} : (\bar{x}, \bar{y}) \in P \}.$$

$$u_i - u_j + n x_{ij} \leq n-1 \quad \forall i \neq 1, \forall j \neq 1,$$

Can be written in matrix form as

$$n I \bar{x} + B \bar{u} \leq (n-1)\, \bar{1} \,, \quad \text{where}$$

$I$ is the identity matrix, $B$ is the arc-node incidence matrix of $G$, with column for node 1 removed, and any non-zero entry corresponding to arcs incident to node 1 zeroed out, and $\bar{1}$ is the vector of ones.



# MATH 567 : Lecture 9 (02/06/2025)

Today: * TSP: MTZ v/s subtour formulations
* sharp formulation of a disjunction

---

Recall: $\text{Proj}_{\bar{x}}(P_{MTZ})$ to compare with $P_{subtour}$

$$u_i - u_j + n x_{ij} \le n-1 \quad \forall i \ne 1, \forall j \ne 1; \longrightarrow \text{matrix form is}$$

$$nI\bar{x} + B\bar{u} \le (n-1)\bar{1} \quad , \quad \text{where}$$

$I$ is the identity matrix, $B$ is the arc-node incidence matrix of $G$, with column for node 1 removed, and any non-zero entry corresponding to arcs incident to node 1 zeroed out, and $\bar{1}$ is the vector of ones.

$$\text{arcs}\left\{\left[\begin{array}{c|c} nI_m & \overbrace{\begin{array}{ccc} & B & \\ \cdots & \underset{i}{+1} \cdots \underset{j}{-1} & \cdots \end{array}}^{\text{nodes, except } 1} \end{array}\right]\right. \begin{bmatrix} \bar{x} \\ \hline \bar{u} \end{bmatrix} \begin{array}{l} \\ \\ u_2, \ldots, u_n \end{array} \le \begin{bmatrix} n-1 \\ n-1 \\ \vdots \\ n-1 \end{bmatrix}$$

The **node-arc incidence matrix** of a directed graph $G=(V,E)$ with $|V|=n$, $|E|=m$ is an $n \times m$ matrix with a row for each node and a column for each arc, with entries in $\{-1,0,1\}$. The column corresponding to arc $(i,j)$ has a $+1$ in row $i$ and a $-1$ in row $j$, and other entries are zero. $B$ above is the transpose of this matrix, with the modifications made as specified.

Also, recall the definition of projection – we went from $A\bar{x} + B\bar{y} \le b$ to the space of $\bar{x}$ variables by eliminating the "unwanted" $\bar{y}$ variables.

Let $C = \{\bar{v} \geq \bar{0} \mid \bar{v}^T B = \bar{0}^T\}$ be the projection cone.

$\bar{v}$?

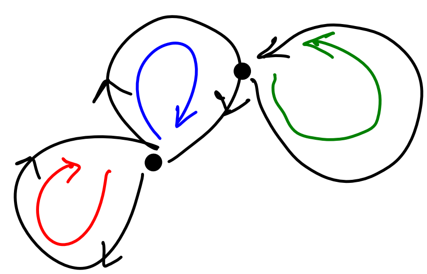
$$C = \{\bar{v} \geq \bar{0} \mid \forall i \neq 1, \underbrace{\sum_j{}' v_{ij} = \sum_j{}' v_{ji}}_{\text{come in and go out at } i \text{ equal \# times}}\}$$

$$= \{\bar{v} \mid \bar{v} \text{ is a } \underline{\text{circulation}} \text{ in } G\}$$

or circuit, generalization of a cycle

$\bar{v}$ is the incidence vector of a circulation.

$\bar{v} \in \{0,1\}^m$, $m = \#$ arcs.



It turns out we can describe all circulations as unions of a finite set of "basic" cycles. In other words, the projection cone is finitely generated.

$$C = \{\sum_{i=1}^{k} \lambda_i \bar{v}^i \mid \lambda_i \geq 0\} \quad \text{where}$$

$\bar{v}^1, \ldots, \bar{v}^k$ are the incidence vectors of a set of basic cycles.

$$\Rightarrow \mathrm{Proj}_{\bar{x}}(P_{MTZ}) = \left\{ \bar{x} \;\middle|\; (\bar{v}^i)^T (n I) \bar{x} \le (\bar{v}^i)^T (n-1) \bar{1},\ i=1,\dots,k; \text{ and system } (1) \right\}.$$

We do not get any inequalities stronger than $\triangle$ here.

If $(\bar{v}^i)^T = [0, 0, \dots, 0, \overbrace{1, 1, \dots, 1}, 0, \dots, 0]$,

1's for $(i,j) \in C$; assumed to be set all together here WLOG.

then

$$(\bar{v}^i)^T (n I) \bar{x} \le (\bar{v}^i)^T (n-1) \bar{1}$$

$$n\, X(C) \le (n-1)\, |C|$$

$$\Rightarrow \quad X(C) \le \left(1 - \tfrac{1}{n}\right) |C|, \text{ which is } \triangle.$$

$\Rightarrow$ The subtour formulation is stronger!

# Sharp Formulation of a Disjunction

Let $S = Q_1 \cup \cdots \cup Q_k$ where $Q_i = \{\bar{x} \mid A_i \bar{x} \le \bar{b}^i\}$, $i=1,\dots,k$.
$\bar{x} \in \mathbb{R}^n$, are non-empty polyhedra with the same recession cone. Then
$$P \subseteq \underbrace{\mathbb{R}^n}_{\bar{x}} \times \underbrace{\mathbb{R}^k}_{\bar{y}} \times \underbrace{(\mathbb{R}^n \times \cdots \times \mathbb{R}^n)}_{k \text{ copies, for } \bar{x}^1,\dots,\bar{x}^k}$$

defined as the set of all vectors $(\bar{x}, \bar{y}, \bar{x}^1, \dots, \bar{x}^k)$ that satisfy

$$P = \left\{ \begin{array}{l} A_1 \bar{x}^1 \le \bar{b}^1 y_1 \\ \quad \vdots \qquad \vdots \\ A_k \bar{x}^k \le \bar{b}^k y_k \\ \bar{x}^1 + \cdots + \bar{x}^k = \bar{x} \\ y_1 + \cdots + y_k = 1 \\ 0 \le y_i \le 1 \ \forall i \end{array} \right\}$$

could ignore, as $y_1 + \cdots + y_k = 1$ implies the same with $y_i \ge 0$.

is a sharp formulation for $S$.

$S$ is the same ($*$) set for which we wrote ($*$- big-M) and ($*$- sharp) formulations.

Q: How do we prove $P$ is indeed a sharp formulation?

To show $P$ is a sharp formulation for $S$, we have to show $\quad \mathrm{Proj}_{\bar{x}}(P) = \mathrm{conv}(S)$.

We need $\text{conv}\left(\bigcup_{i=1}^{k} Q_i\right)$ to be closed, but we'll assume that.

It appears the approach to identify all corner points will not work here.
Could we use another approach?

**Def** An inequality $\bar{a}^T \bar{x} \leq \beta$ is a **valid inequality** for $X \subseteq \mathbb{R}^n$ if $\bar{a}^T \bar{x} \leq \beta \;\; \forall \bar{x} \in X$.

We can try to derive conditions that guarantee an inequality is valid for $\text{conv}(S)$ iff it is valid for $\text{Proj}_{\bar{x}}(P)$.

$$\bar{a}^T \bar{x} \leq \beta \text{ is valid for } \text{conv}(Q_1 \cup \dots \cup Q_k)$$

$$\Longleftrightarrow \bar{a}^T \bar{x} \leq \beta \text{ is valid for each of } Q_1, \dots, Q_k.$$

$$\Longleftrightarrow \exists\, \bar{u}^i \geq \bar{0}, \quad \bar{a}^T = (\bar{u}^i)^T A_i, \quad (\bar{u}^i)^T \bar{b}^i \leq \beta, \text{ i.e.,}$$

we can derive $\bar{a}^T \bar{x} \leq \beta$ from $A_i \bar{x} \leq \bar{b}^i$.

$\bar{a}^T\bar{x} \le \beta$ is valid for $\mathrm{Proj}_{\bar{x}}(P) \iff \exists$ multipliers that derive $\bar{a}^T\bar{x} \le \beta$ from $P$ by eliminating $\bar{x}^1, \ldots, \bar{x}^k, y_1, \ldots, y_k$.

$$
\begin{array}{l|lllllllll}
\bar{a}^T & \bar{x} & -\bar{x}^1 & -\bar{x}^2 & \cdots & -\bar{x}^k & & & & = \bar{0} \\
\bar{v}^{1T} & & A_1\bar{x}^1 & & & & -\bar{b}^1 y_1 & & & \le \bar{0} \\
\bar{v}^{2T} & & & A_2\bar{x}^2 & & & & -\bar{b}^2 y_2 & & \le \bar{0} \\
\vdots & & & & \ddots & & & & & \vdots \\
\bar{v}^{kT} & & & & & A_k\bar{x}^k & & & -\bar{b}^k y_k & \le \bar{0} \\
\beta' \longrightarrow & & & & & & y_1 & + y_2 + \cdots & + y_k & = 1 \\
\beta_1' & & & & & & -y_1 & & & \le 0 \\
\beta_2' & & & & & & & -y_2 & & \le 0 \\
\vdots & & & & & & & \ddots & & \vdots \\
\beta_k' & & & & & & & & -y_k & \le 0
\end{array}
$$

We need

$$\bar{a}^T = (\bar{v}^1)^T A_1 \rightarrow \text{eliminates } \bar{x}^1$$

$$\vdots$$

$$\bar{a}^T = (\bar{v}^k)^T A_k \rightarrow \text{eliminates } \bar{x}^k$$

We need

$\bar{a} \ge \bar{0}$, $\bar{v}^i \ge \bar{0}$,

$\beta' \ge 0$, $\beta_i' \ge 0$

and $\beta' \le \beta$

and

$$(-\bar{v}^1)^T\bar{b}^1 + \beta' - \beta_1' = 0 \rightarrow \text{eliminate } y_1$$

$$\vdots$$

$$(-\bar{v}^k)^T\bar{b}^k + \beta' - \beta_k' = 0 \rightarrow \text{eliminate } y_k$$

$$\left\{\begin{array}{l} (-\bar{v}^1)^T\bar{b}^1 + \beta' \ge 0 \\ \vdots \\ (-\bar{v}^k)^T\bar{b}^k + \beta' \ge 0 \end{array}\right.$$

Notice we need $\beta_i' \geq 0$, and hence could scale the right-hand sides of these inequalities to get rid of $\beta_i'$s.

$$\left\{ \begin{array}{ll} \bar{a}^T = (\bar{v}^i)^T A_i \, , & i = 1, \ldots, k \\ \beta' \geq (\bar{v}^i)^T \bar{b}^i , & i = 1, \ldots, k \\ \beta' \leq \beta & \end{array} \right\}$$

Need to show this system has non-negative solution in $(\bar{v}^i, \beta')$.

We could use this approach for specific instances in which the $A_i$ and $\bar{b}^i$ are provided.

# Definitions and Results on Polyhedra

We collect several relevant definitions and results related to polyhedra here. We will use these results in further elucidating properties and strengths of formulations, as well as comparing them.

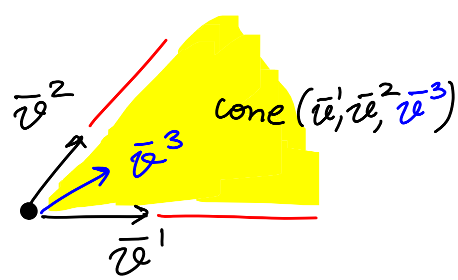
* $C \subseteq \mathbb{R}^n$ is convex if $\lambda\bar{x} + (1-\lambda)\bar{y} \in C \;\forall\, \bar{x}, \bar{y} \in C,\ \lambda \in [0,1]$.

* $C \subseteq \mathbb{R}^n$ is a convex cone if $\lambda\bar{x} + \mu\bar{y} \in C \;\forall\, \bar{x}, \bar{y} \in C$, and $\lambda, \mu \geq 0$.

* $\text{cone}(\{\bar{v}^1, \ldots, \bar{v}^k\}) = \{\bar{x} \mid \bar{x} = \sum_{i=1}^{k} \lambda_i \bar{v}^i,\ \lambda_i \geq 0 \;\forall i\}$.

  the smallest cone containing $\bar{v}^1, \ldots, \bar{v}^k$.

  $k$ is finite $\Longrightarrow$ $C$ is a finitely generated cone.




* A cone $C$ is polyhedral if $C = \{\bar{x} \mid A\bar{x} \leq \bar{0}\}$. Here, $C$ is the intersection of finitely many linear half-spaces. $\{\bar{x} \mid \bar{a}^T\bar{x} \leq 0\}$

* A convex cone is polyhedral iff it is finitely generated.

# MATH 567: Lecture 10 (02/11/2025)

Today: * Definitions on polyhedra
* Integral polyhedra

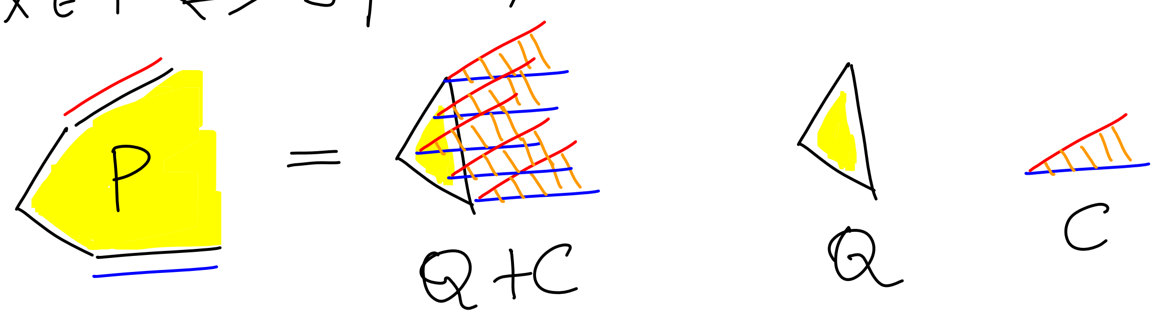
---

* $P \subseteq \mathbb{R}^n$ is a (convex) polyhedron if $P = \{\bar{x} \mid A\bar{x} \leq \bar{b}\}$.
$P$ is the intersection of finitely many affine half-spaces $\{\bar{x} \mid \bar{a}^T\bar{x} \leq \beta\}$, $\bar{a} \neq \bar{0}$, $\beta \neq 0$.
some entries in $\bar{b}$ are $\neq 0$ (not necessary to have all entries $\neq 0$)

* $P \subseteq \mathbb{R}^n$ is a (convex) polytope if it is the convex hull of finitely many vectors.
$P = \text{conv}(\bar{v}^1, \ldots, \bar{v}^k) = \{\bar{x} \mid \bar{x} = \sum_{i=1}^{k} \lambda_i \bar{v}^i, \ 0 \leq \lambda_i \leq 1, \sum_{i=1}^{k} \lambda_i = 1\}$.

$P$ is a bounded polyhedron.

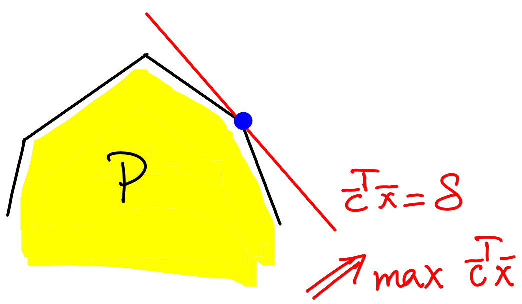
* Motzkin's decomposition theorem: $P$ is a polyhedron iff $P = Q + C$ for some polytope $Q$ and convex cone $C$.
$\bar{x} \in P \iff \exists \bar{y} \in Q, \bar{z} \in C$, s.t. $\bar{x} = \bar{y} + \bar{z}$.




Polytopes and convex cones both have several nice structural properties that might not always hold for general polyhedra. But because of this decomposition theorem, we could present results in terms of polytopes and convex cones.

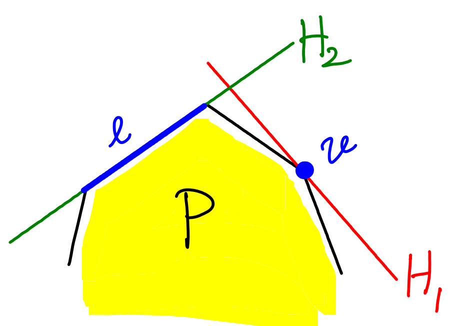
$*$ Farkas' lemma: $\exists\, \bar{x} \mid A\bar{x} \le \bar{b} \iff$
$\nexists\, \bar{u} \ge \bar{0},\ \bar{u}^T A = \bar{0}^T,\ \bar{u}^T \bar{b} < 0.$

$*$ ① Let $P = \{\bar{x} \mid A\bar{x} \le \bar{b}\}$, $\delta = \max\{\bar{c}^T\bar{x} \mid \bar{x} \in P\}$, $\bar{c} \ne \bar{0}$.
Then the affine hyperplane $\{\bar{x} \mid \bar{c}^T\bar{x} = \delta\}$ is a **supporting hyperplane** of $P$.




The line $\bar{c}^T\bar{x} = \delta$ "supports" the polyhedron here.

② $F \subseteq P$ is called a **face** of $P$ if $F = P$ or if $F = P \cap H$ for some supporting hyperplane $H$ of $P$.




Vertex $v$ and line segment $\ell$ are faces of $P$ here

We give an alternative definition for a face of P.

③ If $\bar{a}^T\bar{x} \le \beta$ is a valid inequality for $P$, and $F = \{\bar{x} \in P \mid \bar{a}^T\bar{x} = \beta\}$, then $F$ is a face of $P$.

(4) $F$ is a **proper face** of $P$ if $F \neq \phi$, $F \neq P$, and $F$ is a face of $P$.

$\bar{a}^T\bar{x} \leq \beta$ : $[\bar{a}, \beta]$ represents the face defined by $\bar{a}^T\bar{x} = \beta$.

valid inequality

compact notation

Also $[\bar{a}, \beta]$ supports $P$.

* Alternatively, $F$ is a face of $P \Longleftrightarrow F = \{\bar{x} \mid \bar{x} \in P, A'\bar{x} = \bar{b}'\}$ where $A'x \leq \bar{b}'$ is a subsystem of $A\bar{x} \leq \bar{b}$.

(i) $P$ has only finitely many faces;

(ii) each face is a nonempty polyhedron; and

(iii) if $F$ is a face of $P$, then $F' \subseteq F$ is a face of $P \Longleftrightarrow F'$ is a face of $F$.

* Active (tight) constraint: A constraint $\bar{a}^T\bar{x} \leq \beta$ from $A\bar{x} \leq \bar{b}$ is **tight** or **active** in a face $F$ if $\bar{a}^T\bar{x} = \beta$ $\forall \bar{x} \in F$.

also, binding

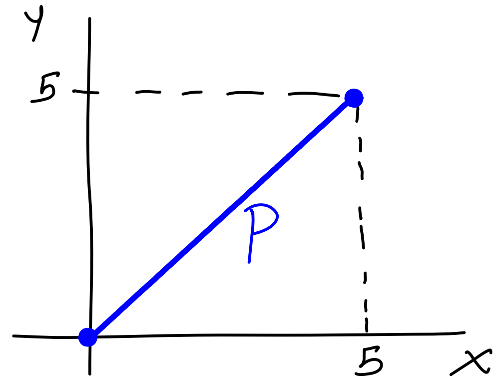
* An inequality $\bar{a}^T\bar{x} \leq \beta$ from $A\bar{x} \leq \bar{b}$ is an **implicit equality** if $A\bar{x} \leq \bar{b} \Longrightarrow \bar{a}^T\bar{x} = \beta$.

* Let $A'\bar{x} \leq \bar{b}'$ be the subsystem of implicit equalities in $A\bar{x} \leq \bar{b}$. Then the **dimension** of $P$ is

$$\dim(P) = n - \operatorname{rank}(A').$$

## Example

$y \geq x$
$y \leq x$
$x \leq 5$
$x \geq 0$

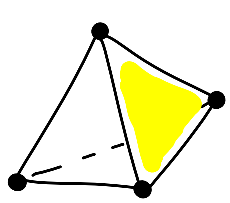



Both $y \geq x$ and $y \leq x$ are implicit equalities here. We get $x - y \leq 0$ and $-x + y \leq 0$, to give $A' = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$, which has $\operatorname{rank}(A') = 1$. Thus, $\dim(P) = 2 - 1 = 1$, which agrees with our intuition.

* $P$ is **full-dimensional** if $\dim(P) = n$, i.e., it has no implicit equalities.

* $\dim\{\bar{v}\} = 0$. (one point/vertex)
* by convention, $\dim(\emptyset) = -1$.
* The **affine hull** of $P$ is $\operatorname{affhull}(P) = \{\bar{x} \mid A'\bar{x} = \bar{b}'\}$.

* **facet**: inclusionwise minimal face $F$ of $P$ with $F \neq P$.
* If $F$ is a facet of $P$, then $\dim(F) = \dim(P) - 1$.



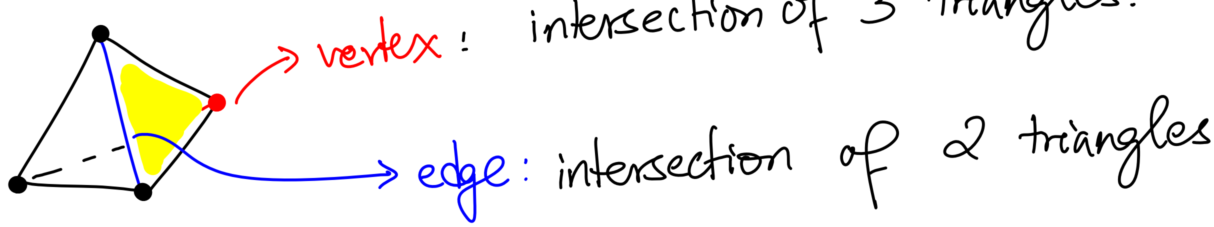
P is a solid tetrahedron.
Each triangle is a facet.
Each vertex and edge is a face,
but not a facet.

Let $A'\bar{x} \le \bar{b}'$ be the subsystem of implicit equalities in $A\bar{x} \le \bar{b}$, and $A^{+}\bar{x} \le \bar{b}^{+}$ be the remaining inequalities.

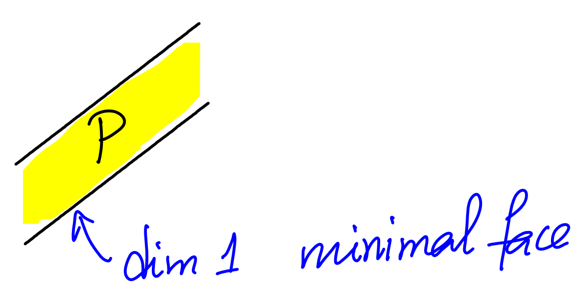
If no inequality in $A^{+}\bar{x} \le \bar{b}^{+}$ is redundant in $A\bar{x} \le \bar{b}$, then for any facet $F$ of $P$,
$F = \{\bar{x} \in P \mid \bar{a}^{+T}\bar{x} = \beta^{+}\}$ for an inequality $\bar{a}^{+T}\bar{x} \le \beta^{+}$ from $A^{+}\bar{x} \le \bar{b}^{+}$.

In this case, $\bar{a}^{+T}\bar{x} = \beta^{+}$ **determines** the facet $F$.

* Each face of $P$, except $P$ itself, is the intersection of facets of $P$.

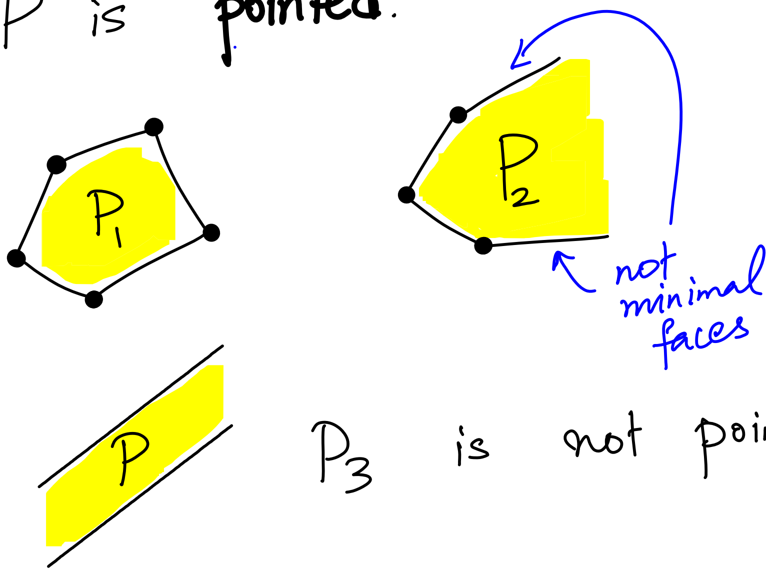



* A **minimal face** of $P$ is a face of $P$ not containing any other face of $P$.
For the tetrahedron, vertices are minimal faces.



* A vertex of $P$ is $\bar{z} \in P$ such that $\{\bar{z}\}$ is a minimal face of dimension zero.

* If each minimal face of $P$ has dimension zero, then $P$ is **pointed**.

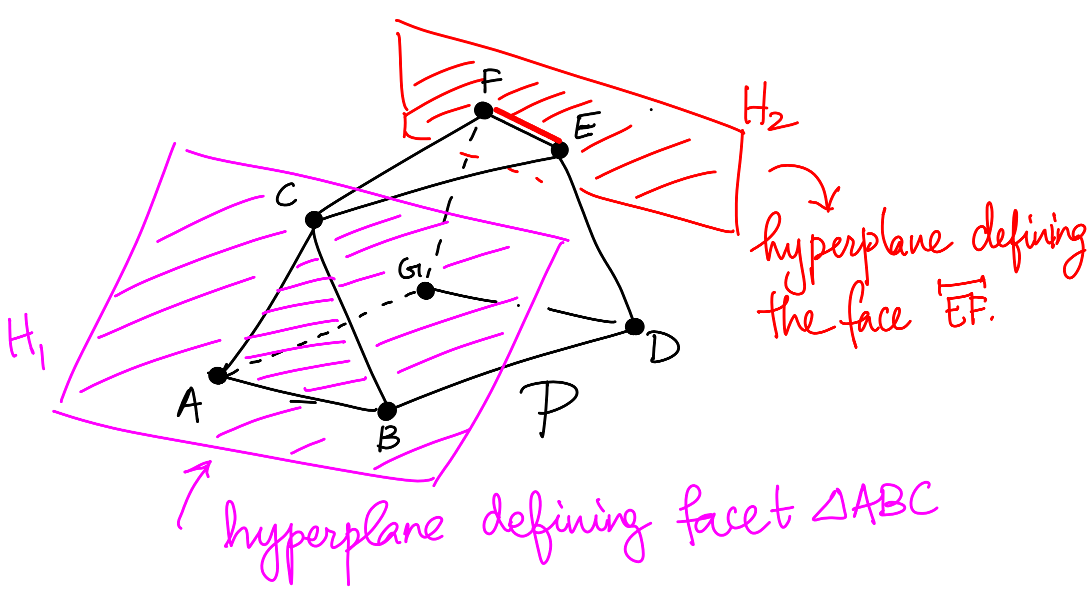



$P_1$ and $P_2$ are pointed.

$P_3$ is not pointed

Here is another example illustrating several of these definitions

$P$ is the solid object in $\mathbb{R}^3$.



$\dim(P) = 3$, and it has no implicit equalities.

All the vertices, edges (line segments), triangles, and quardilaterals are all faces of P.

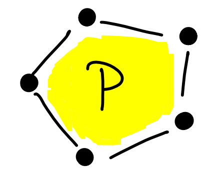
The triangles and quardilaterals are facets of P, and their dimension is 2 each.

The vertices are minimal faces of P. Also, dim(edge) = 1.

$H_1$ is the supporting hyperplane defining face $\triangle ABC$, and $H_2$ defines the face which is edge $\overline{EF}$.

# Special Cases: Well-Solved IPs

Recall: A polytope is the convex hull of its vertices.




We study problems of the form

$$\max\{\bar{c}^T\bar{x} \mid \bar{x} \in X\} = \max\{\bar{c}^T\bar{x} \mid \text{conv}(X)\},$$

where conv(X) is "efficiently" described, i.e., using a polynomial # inequalities in a polynomial # variables. Then we can solve as an LP efficiently (in polynomial time), and get integrality for free.

We restrict our attention to rational polyhedra, i.e., $P = \{\bar{x} \mid A\bar{x} \le \bar{b}\}$ where entries in $A$, $\bar{b}$ are rational.

The subtour formulation with **all** subtour constraints added will describe the convex hull, but there are exponentially many constraints.

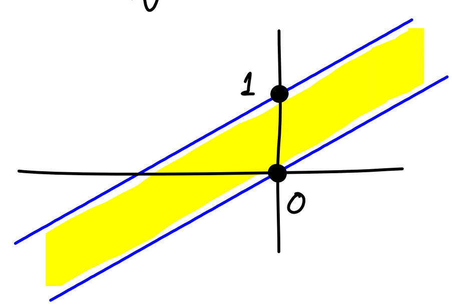
## Integral Polyhedra

**Def** A rational polyhedron is called **integral** if every non-empty face contains an integer vector.

We need to consider only minimal faces.

A pointed rational polyhedron is integral iff **every vertex is integral**.

A polyhedron could be integral even if it is not pointed, e.g., the infinite band as shown here.



# MATH 567: Lecture 11 (02/13/2025)

Today: * unimodularity and total unimodularity

**Recall:** integral polyhedra...

**Theorem 7** (Hoffman 1974) A rational polytope $P$ is integral iff for all integral vectors $\bar{w}$, the optimal value of $\max\{\bar{w}^T\bar{x} \mid \bar{x} \in P\}$ is an integer.

optimal $\bar{w}^T\bar{x}$

**Proof** ($\Rightarrow$) Let $P$ be integral, i.e., all vertices are integral. Then $\max\{\bar{w}^T\bar{x} \mid \bar{x} \in P\}$ is integral for integral $\bar{w}$, as it occurs at a vertex.

($\Leftarrow$) Let $\max\{\bar{w}^T\bar{x} \mid \bar{x} \in P\}$ be integral for all integral $\bar{w}$.

Let $\bar{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$ be a vertex of $P$,

and let $\bar{w}$ be an integral vector such that $\bar{v}$ is the unique optimal solution to $\max\{\bar{w}^T\bar{x} \mid \bar{x} \in P\}$.

We can assume $\bar{w}^T\bar{v} > \bar{w}^T\bar{u} + u_1 - v_1$ for all other vertices $\bar{u}$ of $P$. We can scale $\bar{w}$ by a large integer if needed (i.e., when $u_1 - v_1 < 0$).

Then we have $\bar{w}^T\bar{v} + v_1 > \bar{w}^T\bar{u} + u_1$ for all other vertices $\bar{u}$ of $P$. This inequality gives the following result.

$\Rightarrow$ $\bar{v}$ is the unique optimal solution for the objective function vector $\bar{w}' = [w_1+1, w_2, \ldots, w_n]^T = \bar{w} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$, as

we get $\bar{w}'^T \bar{v} > \bar{w}'^T \bar{u}$ for all other vertices $\bar{u} \in P$.

By assumption, $\bar{w}^T \bar{v}$ and $\bar{w}'^T \bar{v}$ are integral.
Also, $\bar{w}, \bar{w}' \in \mathbb{Z}^n$. $\Rightarrow v_1 \in \mathbb{Z}$.

We repeat the argument for $v_2, v_3, \ldots, v_n$. $\square$

Can extend result to unbounded pointed polyhedra easily, and also to polyhedra in general (i.e., not pointed).

While this theorem specifies an if and only if condition for P to be integral, it does not appear easy to check. We will have to certify integrality of the optimal value for all $\bar{w} \in \mathbb{Z}^n$, for which the optimum exists. But could we specify some easier to check conditions which guarantee integrality?

# Unimodularity and Total Unimodularity (TU)

We assume $A, \bar{b}$ are integral (in $A\bar{x}=\bar{b}$ or $A\bar{x} \le \bar{b}$).

**Def** Let $A \in \mathbb{Z}^{m\times n}$ with full row rank $(\text{rank}(A) = m \le n)$.
$A$ is **unimodular** if each basis of $A$ has determinant $\pm 1$.
$\hookrightarrow B_{m\times m}$ submatrix of $A$ with $\text{rank}(B) = m$.

$A = [B\ N]$, $B \in \mathbb{Z}^{m\times m}$, $\text{rank}(B) = m$, $\det(B) = \pm 1 \Rightarrow B^{-1} \in \mathbb{Z}^{m\times m}$.

For $\bar{b} \in \mathbb{Z}^m$, $A\bar{x} = \bar{b}$ has all integer solutions.

Recall: basic feasible solutions (bfs's) for $\max\{\bar{c}^T\bar{x} \mid A\bar{x}=\bar{b}, \bar{x} \ge \bar{0}\}$ correspond to corner points (vertices) of $\{\bar{x} \mid A\bar{x}=\bar{b}, \bar{x}\ge\bar{0}\}$.

$A = [B\ N]$, $\bar{x} = \begin{bmatrix} \bar{x}_B \\ \bar{x}_N \end{bmatrix}$. Set $\bar{x}_N = \bar{0}$, solve for $\bar{x}_B$.

$$A\bar{x}=\bar{b} \Rightarrow B\bar{x}_B + N\overset{=\bar{0}}{\cancel{\bar{x}_N}} = \bar{b}.$$

We get $\bar{x}_B = B^{-1}\bar{b}$

$\bar{x} = \begin{bmatrix} B^{-1}\bar{b} \\ \bar{0} \end{bmatrix}$ is a basic solution.

If a basic solution $\bar{x}$ satisfies $\bar{x} \ge \bar{0}$, then it's feasible, and hence is a bfs. Each bfs corresponds to a vertex (or corner point) of $P$.

We can use the idea of bfs to give a correspondence between unimodularity of $A$ and integrality of $P$.

**Theorem 8** $A \in \mathbb{Z}^{m \times n}$, rank$(A) = m$. Then $P = \{\bar{x} \mid A\bar{x} = \bar{b}, \underbrace{\bar{x} \geq \bar{0}}_{\text{ensures } P \text{ is pointed}}\}$ is integral for all $\bar{b} \in \mathbb{Z}^m$ iff $A$ is unimodular.

$$A = [B \; N], \quad \bar{x} = \begin{bmatrix} \bar{x}_B \\ \bar{x}_N \end{bmatrix}, \quad \text{set } \bar{x}_N = 0 \Rightarrow \bar{x}_B = B^{-1}\bar{b} \in \mathbb{Z}^m.$$

If $A$ is not unimodular, $\det(B) \neq \pm 1$ for some basis $B$ of $A$, and hence the corresponding basic solution will not be integral for all $\bar{b} \in \mathbb{Z}^m$.

Note that $B^{-1}\bar{b}$ might be integral for some $\bar{b} \in \mathbb{Z}^m$ in this case, e.g., when $|\det(B)| = 2$, and $\bar{b} \in 2\mathbb{Z}^m$. But the result will not hold for all $\bar{b} \in \mathbb{Z}^m$.

We now define a stronger (tighter) property of $A$, which guarantees integrality for polyhedra defined in more general forms. In particular, we would like to relax the requirement that rank$(A) = m$ (i.e., be able to consider more general "shapes" of $A$, e.g., more tall than wide, i.e., $m > n$).

**Def** A matrix $A$ is **totally unimodular** (TU) if every square submatrix of $A$ has determinant $-1, 0,$ or $1$. In particular, $A_{ij} \in \{-1, 0, 1\}$ $\forall i, j$.

e.g., $A = \begin{bmatrix} 3 & 2 \\ 1 & 1 \end{bmatrix}$ is unimodular, but not TU.

$B = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$ is TU (and unimodular here).

$A = \begin{bmatrix} \boxed{\begin{matrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{matrix}} & \begin{matrix} 0 \\ 1 \\ 0 \end{matrix} \\ \begin{matrix} 1 & 1 & 0 \end{matrix} & 1 \end{bmatrix}$ is not TU, as $\det(A_{1:3,1:3}) = -2$.

Notice that we have to check **all** square submatrices in the worst case. But it turns out we could check whether a matrix is TU or not in polynomial time by Seymour's decomposition algorithm, which runs in $O(n^3)$ time. But no implementation is known. Another algorithm by Truemper runs in $O(n^5)$ time, but is implemented, though.

→ check out https://discopt.github.io/cmr/ [Combinatorial Matrix Recognition (CMR) library]

Here is a result that connects total unimodularity and integrality of polyhedra.

**Theorem 9** [Hoffman & Kruskal, 1956]: Let $A \in \mathbb{Z}^{m\times n}$. Then $P = \{\bar{x} \mid A\bar{x} \le \bar{b}, \bar{x} \ge \bar{0}\}$ is integral $\forall \bar{b} \in \mathbb{Z}^m$ for which $P \neq \emptyset$ iff $A$ is totally unimodular (TU).

**Proof** We need a proposition first.

**Proposition** The following statements are equivalent.

(i) $A$ is TU.

(ii) $A^T$ is TU. → determinant is preserved under transposes

(iii) $[A \; I_m]$ is unimodular → full row rank now!

(iv) $\begin{bmatrix} A \\ -A \\ I_n \\ -I_n \end{bmatrix}$ is TU.

Back to proof of Theorem 9.

$P = \{\bar{x} \mid A\bar{x} \le \bar{b}, \bar{x} \ge \bar{0}\}$ is integral $\Longleftrightarrow$

$\text{Proj}_{\bar{x}}(P^z)$ is integral, where $P_z = \{\bar{z} \mid [A \; I]\bar{z} = \bar{b}, \bar{z} \ge \bar{0}\}$

is integral, where $\bar{z} = \begin{bmatrix} \bar{x} \\ \bar{s} \end{bmatrix}$. → slack variables

Now apply Theorem 8, which gives that $P_z$ is integral iff $[A \; I]$ is unimodular, which is true iff $A$ is TU, by the proposition above. $\square$

Here is another characterization of TU and integral polyhedra.

**Theorem 10** Let $A \in \mathbb{Z}^{m\times n}$. $A$ is TU iff $P = \{\bar{x} \mid A\bar{x} \leq \bar{b}\}$ is integral $\forall \bar{b} \in \mathbb{Z}^m$ for which $P \neq \emptyset$.

$\bar{x}$ is urs (unrestricted in sign) here. We can replace $\bar{x}$ by $\bar{x}^+ - \bar{x}^-$, where $\bar{x}^+, \bar{x}^- \geq \bar{0}$, and use Theorem 9.

The upshot is that as long as the constraint matrix of the LP is TU, we're in good shape. The form of the LP—general or standard—does not matter.

## Operations that preserve total unimodularity

1. Swap two rows (or columns).
2. Taking transpose.
3. Scaling a row/column by $-1$.
4. Pivoting, i.e., converting a column to a unit vector using EROs.
5. Adding a zero row/column, or a singleton row/column with the single nonzero entry being $\pm 1$.
6. Repeating a row/column.

⋮

# MATH 567: Lecture 12 (02/18/2025)

Today: * sufficient conditions for TU
* min-cost flow
* LP duality, TDI

---

Some details on the operations that preserve TU...

Example of pivoting:

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \xrightarrow{R_2 - R_3} \begin{bmatrix} \boxed{\begin{matrix} 1 & 1 \\ 1 & -1 \end{matrix}} & \begin{matrix} 0 & 0 \\ 0 & 1 \end{matrix} \\ \begin{matrix} 0 & 1 \\ 1 & 1 \end{matrix} & \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} \end{bmatrix}$$

det = −2

this was the example for a non-TU matrix introduced in Lecture 11...

Equivalently, replacement EROs of the form

$R_i \leftarrow R_i \pm R_j$ preserve TU.

By "preserve", we mean that (when $\tilde{A}$ is obtained by performing the operation on $A$)

$A$ is TU $\Leftrightarrow$ $\tilde{A}$ is TU, and

$A$ is not TU $\Leftrightarrow$ $\tilde{A}$ is not TU.

Seymour's decomposition theorem uses these and a few other operations that preserve TU ($k$-sum, for $k=1,2,3$). It decomposes the task of checking for TU of $A$ into doing the same for several small submatrices obtained by these TU-preserving operations. The TU of these small matrices can be checked immediately (in constant time).

# Sufficient Conditions for TU

**Theorem 11** Let $A \in \{-1,0,1\}^{m\times n}$, with each column having at most one $+1$ and one $-1$. Then $A$ is TU.

**Proof** We prove the result using induction on $k$ for $k\times k$ submatrix $B$ of $A$.

$k=1$. $A_{ij} \in \{-1,0,1\}$. ✓

Induction for $k \geq 2$ (going from $k$ to $k+1$)

If $B$ has a row/column of all zeros, $\det(B)=0$. ✓

If $B$ has a row/column with one nonzero ($\pm 1$), we can expand along that row/column, and using the induction assumption, we get $\det(B) \in \{-1,0,1\}$.

If every column of $B$ has exactly two nonzeros, then adding all rows of $B$ gives the zero vector. Hence $\det(B)=0$.

as every column has one $+1$ and $-1$.

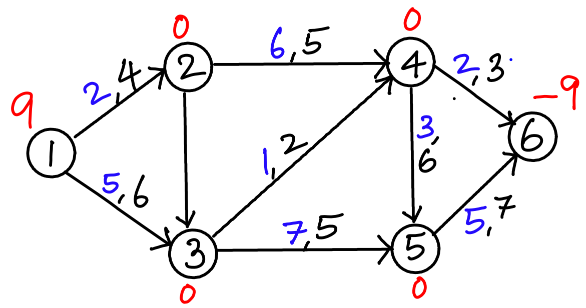
$B\bar{1}=\bar{0}$, where $\bar{1}=\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}$ is a non-trivial solution to $B\bar{x}=\bar{0} \Rightarrow \det(B)=0.$

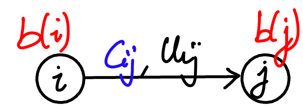
$\square$

# Min-Cost Flow (MCF) on a directed Network

Network matrices satisfy the above sufficient conditions. The node-arc incidence matrix of a directed network (or graph) in the context of the min-cost flow problem is an example.

$G = (V, E)$






$c_{ij}$ : unit cost on $(i,j)$

$u_{ij}$ : capacity (upper bound) of flow on $(i,j)$

Assume: total supply = total demand.

Each node $i$ has supply/demand $b(i)$. If $b(i) > 0$, $i$ is a supply node, and if $b(i) < 0$, $i$ is a demand node. If $b(i) = 0$ then $i$ is a transshipment. The goal is to satisfy demand using the supply by transporting the good through the arcs at the least total cost while honoring arc capacities.

Here is the LP: $x_{ij}$ = flow in arc $(i,j)$.

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij}$$

$$\text{s.t.} \quad \underbrace{\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji}}_{\text{outflow} - \text{inflow}} = \underbrace{b(i)}_{\text{supply/demand}} \quad \forall i \in V$$

$$0 \leq x_{ij} \leq u_{ij} \quad \forall (i,j) \in E.$$

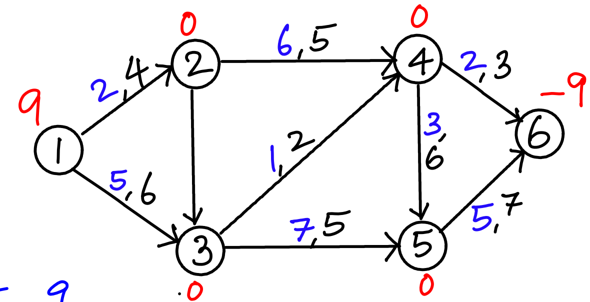
we assume lower bounds are all zero

With $\bar{x} = \begin{bmatrix} \vdots \\ x_{ij} \\ \vdots \end{bmatrix}$, the LP can be written as

$$\begin{aligned} \min \quad & \bar{c}^T \bar{x} \\ \text{s.t.} \quad & A\bar{x} = \bar{b} \\ & I\bar{x} \le \bar{u} \\ & \bar{x} \ge \bar{0} \end{aligned} \qquad \bar{b} = \begin{bmatrix} \vdots \\ b(i) \\ \vdots \end{bmatrix}$$

$$\begin{bmatrix} A \\ I \end{bmatrix} \bar{x} \begin{pmatrix} = \\ \le \end{pmatrix} \begin{bmatrix} \bar{b} \\ \bar{u} \end{bmatrix}$$

where $A$ is the node-arc incidence matrix of $G$, which is guaranteed to be TU as it satisfies the sufficient condition for TU in Theorem 11.




$$\begin{array}{c|ccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ & (1,2) & (1,3) & (2,3) & (2,4) & (3,4) & (3,5) & (4,5) & (4,6) & (5,6) \\ \hline 1 & 1 & 1 & & & & & & & \\ 2 & -1 & & 1 & 1 & & & & & \\ 3 & & -1 & -1 & & 1 & 1 & & & \\ 4 & & & & -1 & -1 & & 1 & 1 & \\ 5 & & & & & & -1 & -1 & & 1 \\ 6 & & & & & & & & -1 & -1 \end{array}$$

$A =$ the bracketed matrix above (rows $1,\dots,6$).

$$\begin{array}{c} \\ i \\ \\ j \end{array} \begin{bmatrix} (i,j) \\ +1 \\ \\ -1 \end{bmatrix}$$

$A$ is TU. From $A$ we get the constraint matrix $\begin{bmatrix} A \\ I \end{bmatrix}$ by adding singleton rows (one for each $(i,j) \in E$), which all preserve TU.

Hence, if $\bar{b}$ and $\bar{u}$ are integral, then the min-cost-flow problem is guaranteed to have integer optimal solutions.

This result does not necessarily hold for undirected graphs.

The same result holds for other problems on directed networks, e.g., shortest path, max flow, transportation, etc. problems. But there are efficient algorithms for each problem — which are faster than solving them as LPs.

We now introduce the concept of total dual integrality, which is a more general concept than TU.

We first do a quick review of LP duality.

# Review of LP Duality

For every linear program (LP), there is another associated LP called its dual LP. Solving the original (primal) LP is equivalent to solving its dual LP, and there are many results relating the two LPs and their interplay.

Here is an example:

(P) primal LP

$$\max\ z = 2x_1 + 3x_2$$

s.t.

$x_1 + 4x_2 \le 5$ $\quad y_1 \ge 0$

$-2x_1 + 3x_2 = 4$ $\quad y_2$ urs

$5x_2 \ge 7$ $\quad y_3 \le 0$

$x_1 \le 0,\ x_2 \ge 0$

$\le \qquad \ge$

(D) dual LP

$$\min\ w = 5y_1 + 4y_2 + 7y_3$$

s.t.

$y_1 - 2y_2 \le 2$

$4y_1 + 3y_2 + 5y_3 \ge 3$

$y_1 \ge 0,\ y_2$ urs, $y_3 \le 0$

Normal vars and normal constraints

max-LP: $\le$ is normal

"maximize revenue s.t. upper bound on raw materials."

min-LP: $\ge$ is normal

"minimize cost s.t. meeting demand, i.e., produce at least a lower bound # units"

$\ge 0$ vars are always normal.

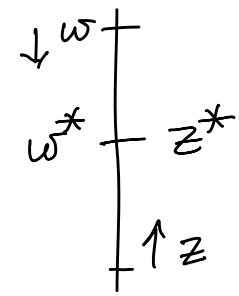
(opposite to) normal vars correspond to (opposite to) normal constraints in the dual LP. And urs vars correspond to $=$ constraints.

# Table of Primal-Dual relationships

| | Primal $\longleftrightarrow$ | Dual |
|---|---|---|
| | min | max |
| variables | $\geq 0$ | $\leq$ → normal |
| | $\leq 0$ | $\geq$ → opposite to normal |
| | urs | $=$ |
| constraints | $\geq$ | $\geq 0$ |
| | $\leq$ | $\leq 0$ |
| | $=$ | urs |

LP duality in matrix form:

$$(P) \quad \begin{array}{ll} \max & z = \bar{c}^T \bar{x} \\ \text{s.t.} & A\bar{x} \leq \bar{b} \quad \bar{y} \geq \bar{0} \end{array} \qquad\qquad \begin{array}{ll} \min & w = \bar{b}^T \bar{y} \\ \text{s.t.} & A^T \bar{y} = \bar{c} \\ & \bar{y} \geq \bar{0} \end{array} \quad (D)$$




One could imagine pushing $z$ up, and pulling $w$ down. Every value of $z$ (i.e., for each feasible solution) lies below every value of $w$. When they are equal, we have optimality for both primal and dual LPs.

## Results on LP duality

1. Weak duality: $z = \bar{c}^T \bar{x} \leq \bar{b}^T \bar{y} = w$ for any feasible $\bar{x}, \bar{y}$ for (P) and (D), respectively.

2. Strong duality: $\bar{c}^T \bar{x} = \bar{b}^T \bar{y} \iff \bar{x}$ and $\bar{y}$ are optimal for (P) and (D), respectively.

The default simplex method tries to push $z$ up to optimality by working on (P). There is an equivalent dual simplex method that pushes $w$ down by working on (D). There are also primal-dual methods which work on both ends, trying to push both $z$ and $w$ to optimality simultaneously.

## Total Dual Integrality (TDI)

LP duality: $\max\{\bar{c}^T\bar{x} \mid A\bar{x} \le \bar{b}\} = \min\{\bar{b}^T\bar{y} \mid A^T\bar{y} = \bar{c},\ \bar{y} \ge \bar{0}\}$ —— (*)

**Def** A system $A\bar{x} \le \bar{b}$ is **totally dual integral** (TDI) if the minimum in (*) is achieved by an integral $\bar{y}$ for each integral $\bar{c}$ for which the optimum exists.

# MATH 567: Lecture 13 (02/20/2025)

Today: * Total Dual Integrality (TDI)
* AMPL

## Total Dual Integrality (TDI) (Recall...)

LP duality: $\max\{\bar{c}^T\bar{x} \mid A\bar{x} \le \bar{b}\} = \min\{\bar{b}^T\bar{y} \mid A^T\bar{y} = \bar{c}, \bar{y} \ge \bar{0}\}$ —— (*)

**Def** A system $A\bar{x} \le \bar{b}$ is **totally dual integral** (TDI) if the minimum in (*) is achieved by an integral $\bar{y}$ for each integral $\bar{c}$ for which the optimum exists.

We present the first result connecting TDI systems and integral polyhedra — it's implication goes only one way, i.e., it is not an "if-and-only-if" result.

**Theorem 12** [Hoffman, 1974] Let $A\bar{x} \le \bar{b}$ be a TDI system such that $P = \{\bar{x} \mid A\bar{x} \le \bar{b}\}$ is a rational polytope and $\bar{b}$ is integral. Then $P$ is an integral polytope.

**Proof** As $\bar{b}$ is integral, and $A\bar{x} \le \bar{b}$ is TDI, $\max\{\bar{c}^T\bar{x} \mid A\bar{x} \le \bar{b}\}$ is integral for all integral $\bar{c}$. Then use Theorem 7.

Note that: TDI is the property of a specific system of inequalities used to describe a polyhedron, and not of the polyhedron itself. So, the same polyhedron could be described by both a TDI system and another system which is not TDI!

## Example 1

$$(P)\quad \begin{array}{lrll} \max & c_1x_1+c_2x_2 & & \\ \text{s.t.} & x_1+x_2 \le b_1 & y_1 \ge 0 \\ & x_2 \le b_2 & y_2 \ge 0 \\ & = \quad = & \end{array}$$

$$(D)\quad \begin{array}{lrl} \min & b_1y_1+b_2y_2 & \\ \text{s.t.} & y_1 & = c_1 \\ & y_1 + y_2 & = c_2 \\ & y_1, y_2 \ge 0 & \end{array}$$

Let $b_1, b_2 \in \mathbb{Z}_{\ge 0}$. For $c_1, c_2 \in \mathbb{Z}$, we solve system in (D) to get

$$\left.\begin{array}{l} y_1 = c_1 \in \mathbb{Z} \\ y_2 = c_2 - c_1 \in \mathbb{Z} \end{array}\right\} \Rightarrow \text{Solution to (D) is integral, when it exists,}$$

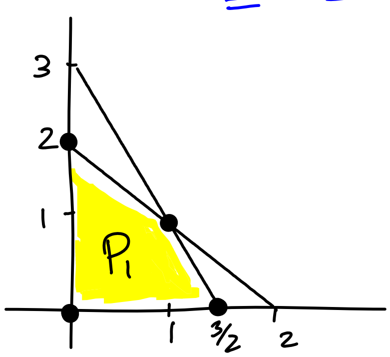
i.e., when $c_1 \ge 0$, $c_2 \ge c_1$. else (D) is infeasible

$\Rightarrow$ The system $\left\{\begin{array}{l} x_1+x_2 \le b_1 \\ x_2 \le b_2 \end{array}\right\}$ is TDI.

## Example 2

$$(P_1)\quad \begin{array}{lrll} \max & z = c_1x_1+c_2x_2 & & \\ \text{s.t.} & x_1+x_2 \le 2 & y_1 \ge 0 \\ & 2x_1+x_2 \le 3 & y_2 \ge 0 \\ & -x_1 \le 0 & y_3 \ge 0 \\ & -x_2 \le 0 & y_4 \ge 0 \\ & = \quad = & \end{array}$$

$$(D)\quad \begin{array}{lrl} \min & w = 2y_1+3y_2 & \\ \text{s.t.} & y_1 + 2y_2 - y_3 & = c_1 \\ & y_1 + y_2 - y_4 & = c_2 \\ & y_i \ge 0 \ \forall i & \end{array}$$




$$\left[\begin{array}{cccc|c} 1 & 2 & -1 & 0 & c_1 \\ 1 & 1 & 0 & -1 & c_2 \end{array}\right] \xrightarrow{R_2-R_1} \left[\begin{array}{cccc|c} 1 & 2 & -1 & 0 & c_1 \\ 0 & -1 & 1 & -1 & c_2-c_1 \end{array}\right]$$

$$\xrightarrow[\text{then } -R_2]{R_1+2R_2} \left[\begin{array}{cccc|c} 1 & 0 & 1 & -2 & 2c_2-c_1 \\ 0 & 1 & -1 & 1 & c_1-c_2 \end{array}\right] \text{ gives}$$

$$\left.\begin{array}{l} y_1 = 2c_2 - c_1 - y_3 + 2y_4 \\ y_2 = c_1 - c_2 + y_3 - y_4 \end{array}\right\} \text{does not help much!}$$

But, for $c_1 = 1, c_2 = 0$, (D) has a unique optimal solution @ $y_2 = y_4 = \frac{1}{2}$, $w^* = \frac{3}{2}$.

Note that $\max\{x_1 \mid \bar{x} \in P_1\} = 3/2$

$P_1$ is not integral!

So, the system describing $(P_1)$ is not TDI!

We get this result also as a contrapositive result to Theorem 12.

It may not be surprising that the polyhedron $(P_1)$ is non-integral and the system describing $(P_1)$ is not TDI. But we could have the reverse case as well — the polyhedron is integral but the system is still not TDI!

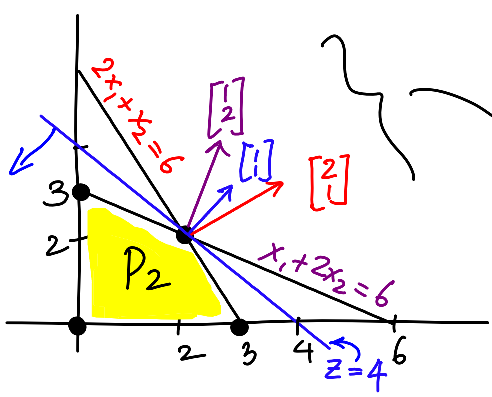
## Example 3

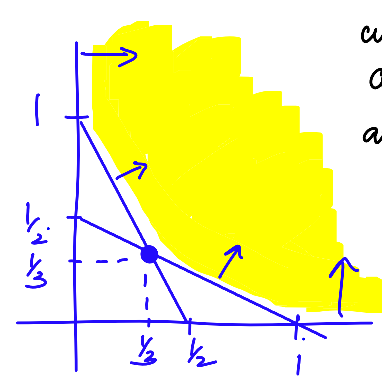
$$(P_2)\quad \begin{array}{ll} \max & z = x_1 + x_2 \\ \text{s.t.} & \left\{ \begin{array}{l} x_1 + 2x_2 \le 6 \\ 2x_1 + x_2 \le 6 \\ x_1, x_2 \ge 0 \end{array} \right\} \begin{array}{l} y_1 \ge 0 \\ y_2 \ge 0 \end{array} \\ & \quad\ \ \ \ge \quad \ge \end{array}$$

$z^* = 4$ at $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$

$$\begin{array}{ll} \min & w = 6y_1 + 6y_2 \\ \text{s.t.} & y_1 + 2y_2 \ge 1 \\ & 2y_1 + y_2 \ge 1 \\ & y_1, y_2 \ge 0 \end{array} \quad (D)$$

$w^* = 4$ at $\begin{bmatrix} 1/3 \\ 1/3 \end{bmatrix}$



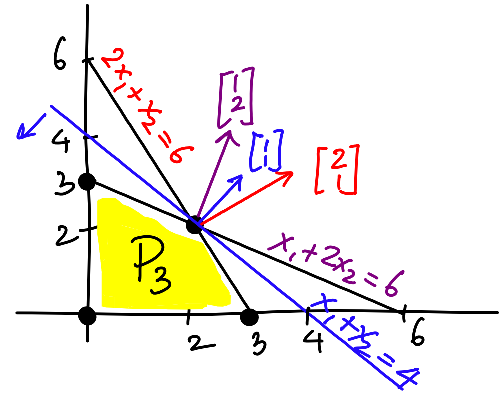

We could treat $x_i \ge 0$ as regular inequalities, use $y_3, y_4$ for them, and still get $y_1 = y_2 = \frac{1}{3}$ as the unique optimal solution!

Normal vectors of the constraints and $z = x_1 + x_2$.

We should be able to express $\begin{bmatrix}1\\1\end{bmatrix}$ as an integer linear combination of $\begin{bmatrix}1\\2\end{bmatrix}$ and $\begin{bmatrix}2\\1\end{bmatrix}$, for an integer (optimal) solution to exist. Hence $(P_2)$ is not TDI.

So, $(P_2)$ is not TDI, even though polytope is integral.

But we can describe $(P_2)$ (the polytope) by another system of inequalities $(P_3)$, which is indeed TDI.




$$\max\ z = x_1+x_2$$
$$\text{s.t.}\quad (P_3)\left\{\begin{array}{l} x_1+2x_2 \le 6 \\ 2x_1+x_2 \le 6 \\ x_1+x_2 \le 4 \\ x_1, x_2 \ge 0 \end{array}\right\} y_3$$

$y_3=1$, $y_1=y_2=0$ is an integral optimal solution to (D), showing is TDI.

Now, $\begin{bmatrix}1\\1\end{bmatrix}$ can indeed be expressed as an integer linear combination of $\begin{bmatrix}1\\2\end{bmatrix}, \begin{bmatrix}2\\1\end{bmatrix}$, and $\begin{bmatrix}1\\1\end{bmatrix}$.

The power of TDI lies oftentimes more on the mathematical side than on the computational/practical side. Knowing that a polyhedron can be described by a TDI system could be useful in proving certain related results.

<u>Theorem 8.13 (Bertsimas-Weismantel)</u>

Every rational polyhedron $P$ can be described as a TDI system of the form $A\bar{x} \le \bar{b}$ with $A$ integral.

<u>Corollary</u> A rational polyhedron $P$ is integral iff there exists a TDI system describing $P$ of the form $A\bar{x}\le\bar{b}$ with $A, \bar{b}$ integral.

# AMPL

See AMPL handout posted on the course web page.

For the Farmer Jones LP (used as the first example), one could use $n$ for the # crops in place of a set of crops. See the course web page for AMPL files using # crops.

## Integer programming example

Knapsack feasibility problem: $\beta' \le \bar{a}^T \bar{x} \le \beta$

$\bar{x} \in \{0,1\}^n$

$a_1 x_1 + \cdots + a_n x_n$

goal is to check feasibility: $\overset{?}{\exists}$ $\bar{x} \in \{0,1\}^n$ satisfying the knapsack inequalities?

There is no objective function (or, one could use a dummy objective function). The goal is to find an integer feasible point $\bar{x}$ satisfying the knapsack bounds, or prove there are no integer feasible solutions. Indeed, the latter case represents the worst case instances for most IP algorithms.

For the instance illustrated in class, we had $n=50$, and all the numbers (50 $a_i$'s, $\beta'$, and $\beta$) were available in a text file. The data could be read into ampl using the **read** command.

# MATH 567: Lecture 14 (02/25/2025)

Today: * Branch-and-Bound (B&B)

## Branch-and-Bound (B&B)

We describe a generic branch-and-bound algorithm for the problem of finding $z(S) = \max\{\bar{c}^T\bar{x} \mid \bar{x} \in S\}$. $S$ here need not be a polyhedron, or disjoint collections of polyhedra, or even polyhedra intersected with $\mathbb{Z}^n$. It could be quite general.

We assume that

* we can divide a subproblem $T \subseteq S$, and
* we can compute lower and upper bounds

$$z_\ell(T) \leq z(T) \leq z_u(T)$$

for $z(T) = \max\{\bar{c}^T\bar{x} \mid \bar{x} \in T\}$.

We should be able to compute these bounds easily, i.e., in polynomial time. For MIP/IP, we usually solve LP relaxations, i.e., the problems without integrality restrictions.

We will describe how to maintain and update these bounds so as to arrive at the final answer.

Here is the generic algorithm (for $z = \max\{\bar{c}^T\bar{x} \mid \bar{x} \in S\}$)

**Step 0** Let $\mathcal{L} = \{S\}$. (the list of (sub) problems).
compute $z_\ell(S), z_u(S)$.

**Step r** (i) Remove a subproblem $T \in \mathcal{L}$. $(\mathcal{L} \leftarrow \mathcal{L}/\{T\})$
(ii) Divide $T$ as $T = T_1 \cup \dots \cup T_k$;
compute $z_\ell(T_i)$, $z_u(T_i)$, $i = 1, \dots, k$.
set $\mathcal{L} = \mathcal{L} \cup \{T_1, \dots, T_k\}$.
(iii) Let $z_\ell(S) = \max\{z_\ell(S), \max\{z_\ell(T) \mid T \in \mathcal{L}\}\}$
save the solution $\bar{x}$ here
integer feasible
(iv) Prune all $T \in \mathcal{L}$ with
$z_u(T) \leq z_\ell(S)$;
throw away; i.e., remove from $\mathcal{L}$
(v) if $\mathcal{L} = \phi$ STOP;
else
set $z_u(S) = \max\{z_u(T) \mid T \in \mathcal{L}\}$;
end

# Correctness of the generic B&B algorithm

**Claim 1** At any time

$$\{\bar{x} \in S' \mid \bar{c}^T\bar{x} > z_\ell(S)\} \subseteq \mathcal{L}.$$

* true in the beginning
* maintained in Step (iv), where we prune a problem from $\mathcal{L}$.

remove

**Claim 2** Update in Step (v) is correct.

Case 1: $z(S) > z_\ell(S)$

If $\bar{x}^*$ has $\bar{c}^T\bar{x}^* = z(S)$, then by Claim 1, $\bar{x}^* \in T$ for $T \in \mathcal{L}$. Then $z_u(T) \geq \bar{c}^T\bar{x}^* = z(S)$.

Case 2: $z(S) = z_\ell(S)$.

Since we are already past Step (iv), we must have

$$z_u(T) > z_\ell(S) = z(S) \quad \forall T \in \mathcal{L}.$$

□

Let's compare the updates of $z_\ell, z_u$:

consider current best value, and those from all children

$$z_\ell(S) = \max\{z_\ell(S), \max\{z_\ell(T) \mid T \in \mathcal{L}\}\}.$$

$$z_u(S) = \max\{z_u(T) \mid T \in \mathcal{L}\}. \longrightarrow \text{consider values only from the children}$$

## More specific details for $k=2$ (we create two subproblems: $S = S_1 \cup S_2$)

1. Partition $S$: let $x_i \in \mathbb{Z}$ in $S$, but $x_i = k+\delta$ in the LP solution for $S$, where $k \in \mathbb{Z}$, $0 < \delta < 1$. Then we can set $S_1 = S \cap \{\bar{x} \mid x_i \leq k\}$ and $S_2 = S \cap \{\bar{x} \mid x_i \geq k+1\}$.

2. Find $z_{uj}$, $j=1,2$ by solving the LP relaxations of $\max\{\bar{c}^T\bar{x} \mid \bar{x} \in S_j\}_{j=1,2}$. If infeasible, set $z_{uj} = -\infty$.

3. Find $z_{\ell j}$, $j=1,2$. Try to find any integer feasible solution $\bar{x}^j \in S_j$, $j=1,2$. If we cannot find an integer solution $\bar{x}^j$, set $z_{\ell j} = -\infty$.

We will have

$$\max_j \{z_j\} = z \qquad \text{——— (1)}$$

$$\max_j \{z_{\ell j}\} \leq z \leq \max_j \{z_{uj}\} \qquad \text{——— (2)}$$

We keep updating the lower and upper bounds by taking the max in each case.

In general, $z_{\ell j}$ comes from an integer feasible solution, and $z_{uj}$ comes from a **relaxation**. The typical relaxation involves relaxing, i.e., ignoring the integrality constraints. But one could ignore any subset of constraints.

Another example of a relaxation:

$S = \{\bar{x} \mid \bar{x}$ is the incidence vector of a TSP tour$\}$.

To get a relaxation of $S$, throw away the subtour constraints.

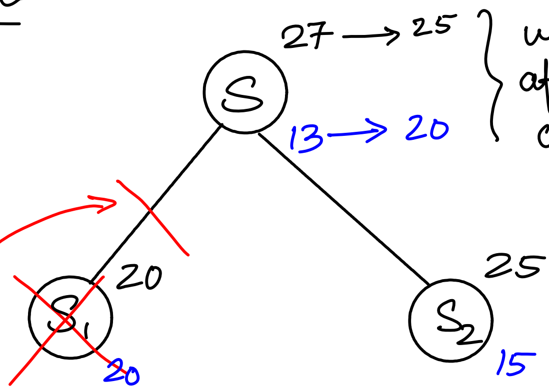
# How do we use (2): update of bounds

## Three examples

$z_{ui}$ $S_i$ $z_{\ell i}$

1.

Notation for pruning: X or

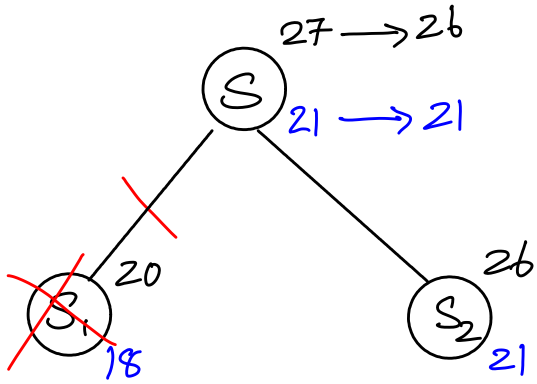



} update bounds after handling child nodes

prune $S_1$ by **optimality**.

We have equality of the bounds in $S_1$. We cannot improve the solution any more, so can prune it. Also, there is no need to branch any more.

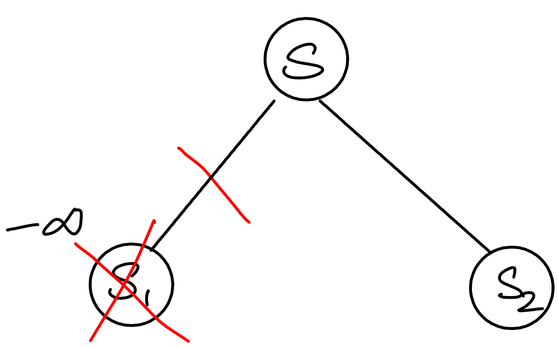
2.




prune $S_1$ by **bound**.

$z_{u1} = 20 < z_\ell = 21$. We cannot possibly find a better solution by branching on $S_1$ any more. So we prune it.

3.



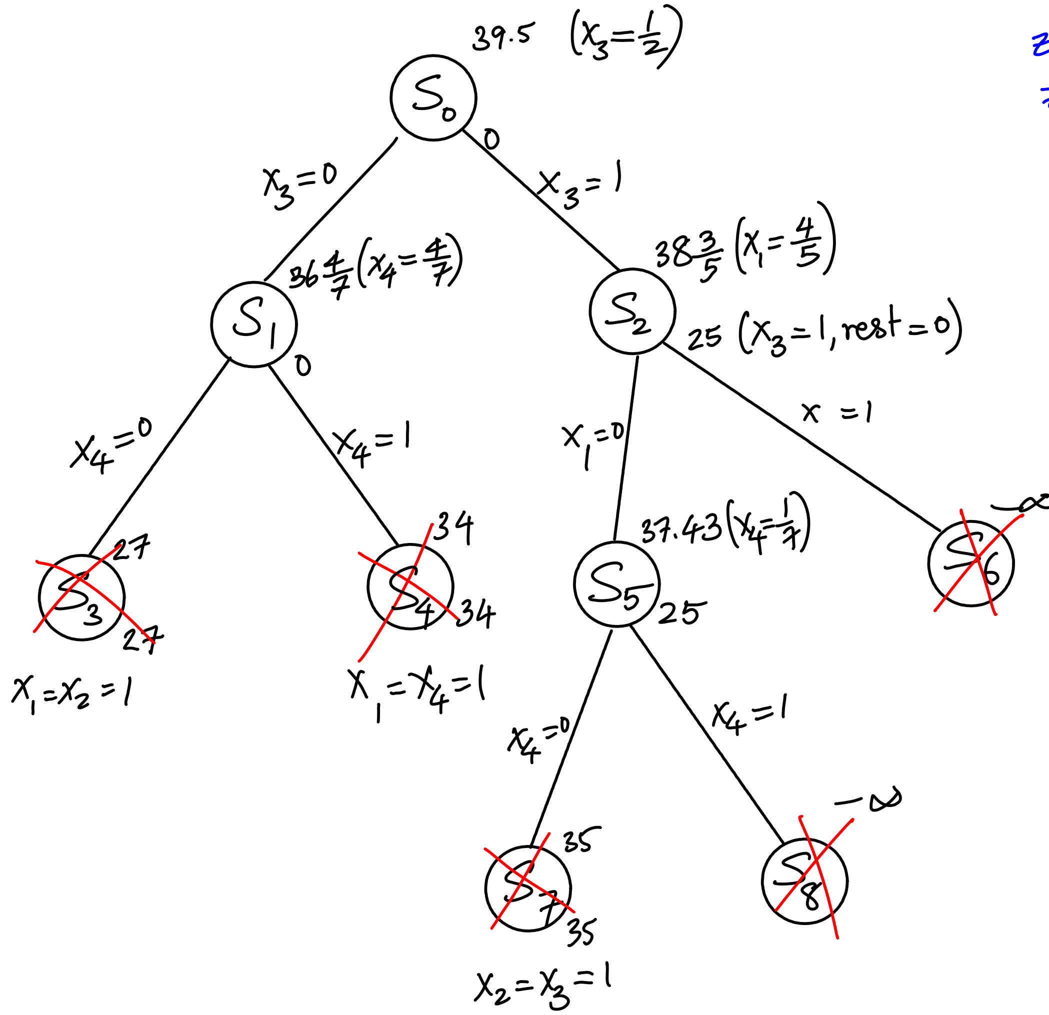

prune $S_1$ by **infeasibility**.

$S_1$ is not feasible even as an LP – no point considering it any further.

# Illustration of B&B on $n=4$ knapsack problem

## Prob 3, Chap 7 of Wolsey, Integer programming

$$\max \ z = 17x_1 + 10x_2 + 25x_3 + 17x_4$$
$$\text{s.t.} \quad 5x_1 + 3x_2 + 8x_3 + 7x_4 \leq 12$$
$$\bar{x} \in \{0,1\}^4$$




$z_u = 39.5$
$z_\ell = 0$

$z_u = 38.6$
$z_\ell = 25$

$z_u = 37.43$
$z_\ell = 34$

$z_u = 35$
$z_\ell = 35$

$S_3, S_4, S_7$ are pruned by optimality, while $S_6$ and $S_8$ are pruned by infeasibility. No nodes are pruned by bound here. See the AMPL session for details.

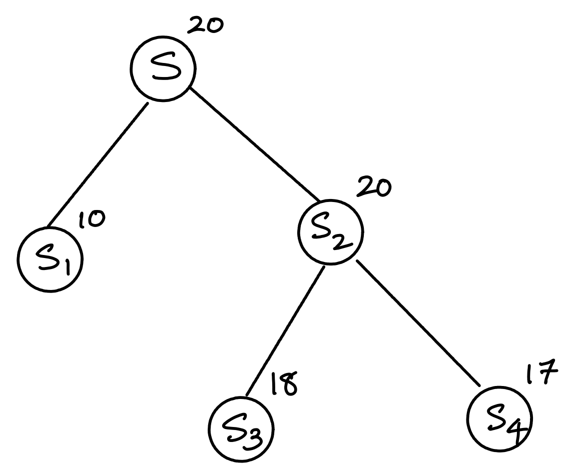
# MATH 567: Lecture 15 (02/27/2025)

Today:
* B&B strategies
* reduced cost fixing in B&B
* types of branching

## Node selection Strategies

How do we select a subproblem from $\mathcal{L}$ ?

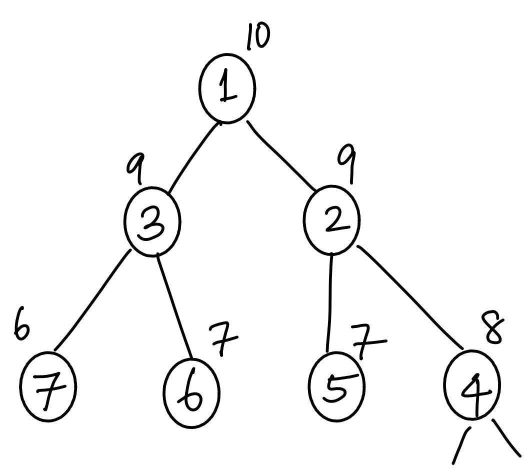
Consider this example:




If we subdivide $S_1$, $z_u$ will remain at 20. If $z^*=15$ (optimal objective function value) and we knew it, we would not subdivide $S_1$. On the other hand, subdividing $S_2$ decreases $z_u$ to 18.

This example seems to suggest that choosing a node with a high $z_u$ may be a good idea. This strategy is called **best node first** (BNF) strategy. (pick subproblem from $\mathcal{L}$ with largest $z_u$).

Another typical BNF B&B tree




nodes are numbered in the order they are examined here

## Advantages of BNF Strategy

* Rapidly decreases $z_u$. → global upper bound
* Never subdivides a node $T_k$ with $z_u(T_k) < z^*$, (→ optimal z-value)
  can prune many nodes, and hence the # nodes to prove optimality is relatively small.

  ↳ assuming you already identified the optimal solution — you still have to prove it is indeed optimal.

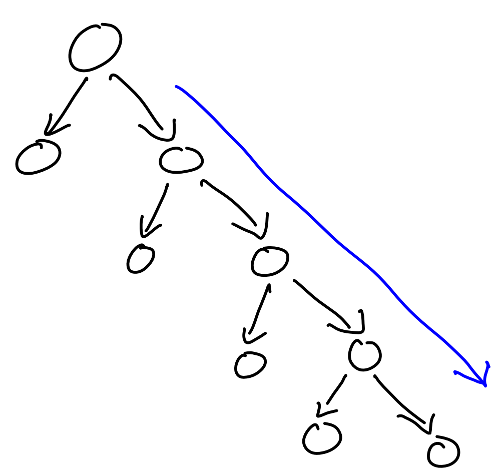
## Disadvantages of BNF strategy

* The B&B tree is widespread, and memory needed to store the list of subproblems $\mathcal{L}$ may be huge.
* May take a long time to find an integer feasible solution (i.e., a node $T_k$ with $z_u(T_k) = z_\ell(T_k)$).

## Depth-First Search (DFS) B&B Strategy

Exact opposite to BNF → always select the problem that was added to $\mathcal{L}$ the last (LIFO order).

A typical DFS B&B tree:



## Advantages of DFS B&B Strategy

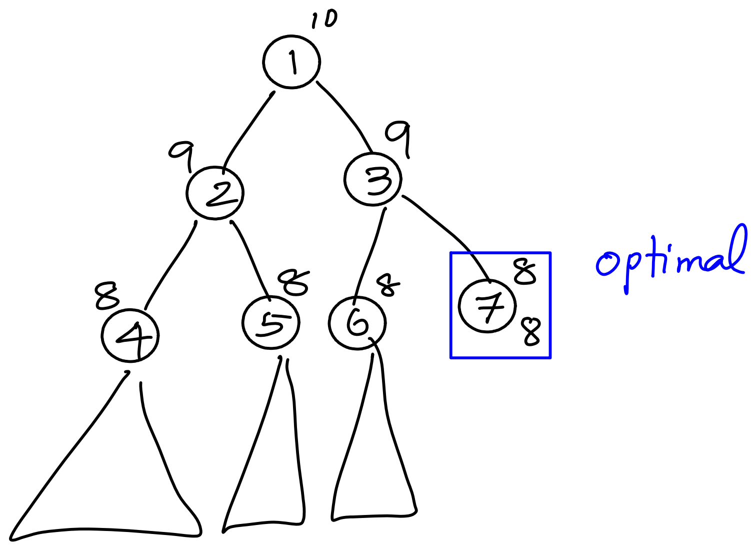
* Maximum depth of B&B tree is $n$; at any point, DFS stores at most $2n$ subproblems in $\mathcal{L}$.
* Since it gets down deep quickly in the B&B tree, DFS find integer feasible solutions relatively quickly.

## Disadvantages of DFS

* If it hits a "wrong" subtree, it may not find a feasible solution, or even change the bounds for a long time.
* It may take a long time to prove optimality.

Let's consider a sample B&B tree, and how both strategies (BNF and DFS) perform on the same. We will consider both "good" and "bad" extremes for their performances — "lucky" or "unlucky"!

## An Example



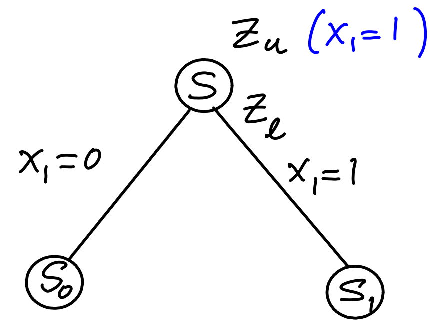

| Strategy | # nodes until finding optimal solution | # nodes until finishing (proving optimality) |
|---|---|---|
| BNF lucky | 4 (1-2-3-7) | 7 (1-2-3-7-4-5-6) |
| BNF unlucky | 7 (1-2-3-4-5-6-7) | 7 (1-2-3-4-5-6-7) |
| DFS lucky | 3 (1-3-7) | 7 (1-3-7-6-2-5-4) |
| DFS unlucky | M | M |

So, DFS is a gambler, while BNF is conservative.

In practice, we combine the two strategies, along with other "intelligent" strategies specific to the problem in hand.

# Reduced Cost fixing in B&B

Consider a 0-1 IP.




Say we solve the LP relaxation at $S$ (to get $z_u(S)$), and in the optimal solution $\bar{x}$, we have $x_1 = 1$. Can we conclude that

$z_u(S_0)$ = LP optimum at $S_0 \leq z_\ell$ ?

If yes, we can fix $x_1 = 1$ (in the optimal solution of IP).

**LP relaxation at S**

$$z_u(S) = \left\{ \begin{array}{lll} \max & \bar{c}^T \bar{x} & \\ \text{s.t.} & A\bar{x} \leq \bar{b} & \rightarrow \bar{y} \geq \bar{0} \\ & \bar{x} \leq \bar{1} & \rightarrow \bar{u} \geq \bar{0} \\ & -\bar{x} \leq \bar{0} & \rightarrow \bar{v} \geq \bar{0} \end{array} \right. \quad (P)$$

$=$

(P) ↑ primal LP

**Dual**

$$\left. \begin{array}{ll} \min & \bar{b}^T \bar{y} + \bar{1}^T \bar{u} \\ \text{s.t.} & A^T \bar{y} + \bar{u} - \bar{v} = \bar{c} \\ & \bar{y}, \bar{u}, \bar{v} \geq \bar{0} \end{array} \right\} (D)$$

(D) ↑ dual LP

**Theorem 13** Suppose the optimal solution to (P) and (D) be $(\bar{x}, \bar{y}, \bar{u}, \bar{v})$ with

1. $x_1 = 1$, and
2. $u_1 \geq z_u(S) - z_\ell$

also = opt (D), the optimal obj. fn of dual (D)

Then $z_u(S_0) \leq z_\ell$. So we can fix $x_1 = 1$.

Recall complementary slackness conditions — if a constraint in (P) is satisfied as a strict inequality, i.e., it's nonbinding, the corresponding dual variable will be zero in the optimal solution. So, $v_1 = 0$ here (as $-x_1 \leq 0$ is not binding).

**Proof**

Consider the dual LP at $S_0$. $(D) \wedge (x_1 = 0)$ is the same as $(D)$, but with $v_1$ free (urs). $v_1$ appears in $(D)$ only in $(A^T\bar{y})_1 + u_1 - v_1 = c_1$. Hence a feasible solution to $(D) \wedge (x_1=0)$ (i.e., $(D)$ at $S_0$) is given by $(\bar{y}', \bar{u}', \bar{v}')$, where

$$\bar{y}' = \bar{y}, \quad \bar{u}' = \bar{u}, \quad \bar{v}' = \bar{v} \quad \text{except for } u_1' = 0, \; v_1' = -u_1.$$

$\Rightarrow$ The optimal obj. fn. value at $(D) \wedge (x_1 = 0) \leq z_u(S) - u_1$.

$$\leq z_\ell \text{ by } (2).$$

Hence we can prune $S_0$, i.e., fix $x_1 = 1$. $\square$

In practice, reduced cost fixing and other similar strategies are all implemented as part of B&B (for example, in CPLEX). In fact, packages such as CPLEX do much more than simple B&B. Still, there are some pathological instances of certain IPs, which are bad for CPLEX even at moderate dimensions ($\leq 100$!).

## Example

$$(P)\left\{\begin{array}{lll} \max & 2x_1 + x_2 & \\ \text{s.t.} & x_1 + 2x_2 \leq 2 & y \\ & x_1 \leq 1 & u_1 \\ & x_2 \leq 1 & u_2 \\ & -x_1 \leq 0 & v_1 \\ & -x_2 \leq 0 & v_2 \end{array}\right.$$

$$\begin{array}{ll} \min & 2y + u_1 + u_2 \\ \text{s.t.} & y + u_1 - v_1 = 2 \\ & 2y + u_2 - v_2 = 1 \\ & y, u_1, u_2, v_1, v_2 \geq 0 \end{array} \quad (D)$$

$z_\ell = z^* = 2$ with $\bar{x}^* = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$.

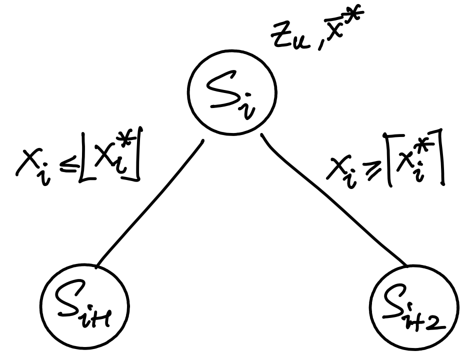
$z_u = 5/2$ with $\bar{x} = \begin{bmatrix} 1 \\ 1/2 \end{bmatrix}$.

For $(D)$, opt. solution is $y = \frac{1}{2}$, $u_1 = 3/2$.

$u_1 \geq z_u - z_\ell = \frac{5}{2} - 2 = \frac{1}{2}$.

So, fix $x_1 = 1$ by Theorem 13.

# Types of Branching

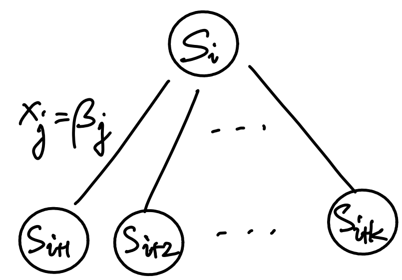



## ① Binary branching

Solve LP relaxation at $S_i$.
Let the optimal solution to this LP relaxation be $\bar{x}^*$ with $x_i^*$ non-integral, where $x_i \in \mathbb{Z}$ is required. Create two branches by adding $x_i \le \lfloor x_i^* \rfloor$ and $x_i \ge \lceil x_i^* \rceil$.

Example: $x_5 = 13.6$ (in $\bar{x}^*$). Create the branches $x_5 \le 13$ and $x_5 \ge 14$.

Binary variables are indeed covered in this case.




$k = \lfloor \gamma_{ij} \rfloor - \lceil \delta_{ij} \rceil + 1$

## ② Integer branching

Choose a variable $x_j$ that needs to be integral, find

$$\delta_{ij} = \min \{ x_j \mid \bar{x} \in \text{LP relaxation at } S_i \},$$
$$\gamma_{ij} = \max \{ x_j \mid \bar{x} \in \text{LP relaxation at } S_i \}.$$

Create branches by adding constraints

$$x_j = \beta_j \quad \text{where} \quad \beta_j \in \{ \lceil \delta_{ij} \rceil, \lceil \delta_{ij} \rceil + 1, \dots, \lfloor \gamma_{ij} \rfloor \}.$$

So, create $\lfloor \gamma_{ij} \rfloor - \lceil \delta_{ij} \rceil + 1$ nodes.

e.g., $\delta_{ij} = 13.64$, $\gamma_{ij} = 16.39$ for $\delta_{ij} \le x_j \le \gamma_{ij}$; we create 3 branches with $x_j = 14, 15, 16$.

# MATH 567 : Lecture 16 (03/04/2025)

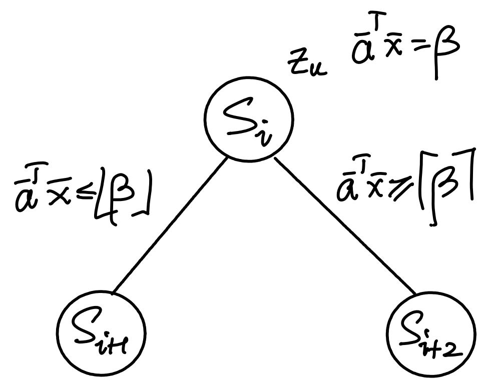
Today: * branching on constraint
* Jeroslow's IP
* cutting planes

---

## Types of Branching (continued..)

### ③ Binary branching on a constraint:

Assume IP, i.e., $\bar{x} \in \mathbb{Z}^n$ is required.
Let $\bar{a}^T\bar{x} = \beta$ is valid for the LP relaxation at $S_i$, where $\beta$ is non-integral, but $\bar{a} \in \mathbb{Z}^n$. We then create two nodes by adding $\bar{a}^T\bar{x} \leq \lfloor \beta \rfloor$ and $\bar{a}^T\bar{x} \geq \lceil \beta \rceil$.




<u>Example</u> Let $2x_1 + 3x_2 + 5x_3 = 7.43$ hold for LP at $S_i$, where $x_1, x_2, x_3 \in \mathbb{Z}$. Create two branches by adding $2x_1 + 3x_2 + 5x_3 \leq 7$ and $2x_1 + 3x_2 + 5x_3 \geq 8$.

### ④ Integer branching on a constraint:

Similar to ②, but for a constraint $\bar{a}^T\bar{x}$, $\bar{a} \in \mathbb{Z}^n$.

<u>Example</u> Let $6.71 \leq 3x_1 + 5x_3 - x_4 + 2x_5 \leq 11.99$ be valid, where $x_1, x_3, x_4, x_5 \in \mathbb{Z}$ is required. We create five branches by adding $3x_1 + 5x_3 - x_4 + 2x_5 = \beta$ for $\beta = 7, 8, 9, 10, 11$ $\left(7 = \lceil 6.71 \rceil,\ 11 = \lfloor 11.99 \rfloor\right)$.

## Jeroslow's IP (1974)

$$\begin{aligned} \min \quad & x_{n+1} \\ \text{s.t.} \quad & 2x_1 + 2x_2 + \cdots + 2x_n + x_{n+1} = n \quad \text{for **odd** } n \\ & x_j \in \{0,1\}, \; j = 1,2,\ldots,n+1. \end{aligned}$$

The optimal solution must set $x_{n+1} = 1$. But binary branching on variables (option ①) will take an exponential number (in $n$) of nodes to solve it!

## Feasibility version of Jeroslow's IP

We will prove the above result for this simpler version.

$n = 2k+1$ (odd). Consider the following feasibility binary IP:

$$\begin{aligned} 2x_1 + 2x_2 + \cdots + 2x_n &= 2k+1 \\ x_j \in \{0,1\}, \; j &= 1,2,\ldots,n. \end{aligned}$$

The goal here is to **prove** that the above IP is integer infeasible using B&B.

Say, $x_1, \ldots, x_j$ for $j < k$ are fixed already (WLOG). Also, assume $x_r = 1$ for $r = 1, \ldots, i$ for $i < j$, and $x_r = 0$ for $r = i+1, \ldots, j$.

The LP feasibility problem at the current node is

$$\begin{aligned} 2x_{j+1} + \cdots + 2x_{2k+1} &= 2k+1-2i. \\ 0 \le x_r \le 1, \; r &= j+1, \ldots, 2k+1. \end{aligned}$$

$$\Rightarrow \quad \underbrace{x_{j+1} + \cdots + x_{2k+1}}_{2k+1-j} = \frac{2k+1-2i}{2} = k - i + \frac{1}{2}$$

$$0 \le x_r \le 1, \ r = j+1, \ldots, 2k+1.$$

As long as $j < k$, $2k+1-j > k+1$. So we can always find an LP-feasible solution (non-integral) to this subproblem. Hence we cannot prune this node! In fact, there may be many LP-feasible solutions.

$\Rightarrow$ We have to fix at least $k = \lfloor \frac{n}{2} \rfloor$ of the $x_i$'s before we can prune a node. Hence the B&B tree has at least $2^k = 2^{\lfloor n/2 \rfloor}$ nodes.

But we could prove integer infeasibility at the root node itself by branching on the constraint $x_1 + x_2 + \cdots + x_n$!

$$\max/\min \ \sum_{j=1}^{n} x_j$$

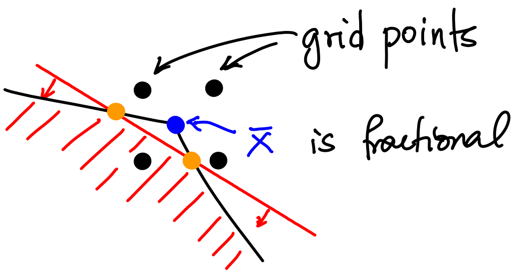
$$\text{s.t.} \quad 2\sum_{j=1}^{n} x_j = 2k+1$$

$$0 \le x_j \le 1, \ j = 1, 2, \ldots, 2k+1.$$

$\delta(\min) = \gamma(\max) = k + \frac{1}{2}$, and hence $\lceil \delta \rceil > \lfloor \gamma \rfloor$.

So we create zero nodes!

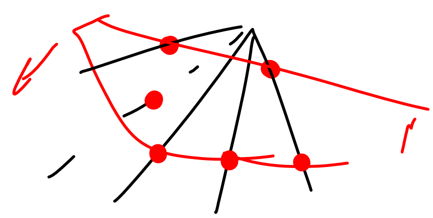
# Cutting Planes



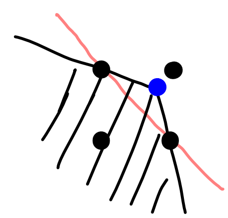

A cutting plane cuts off a non-integral corner point (in the feasible region of the LP relaxation).

As illustrated here, cutting a fractional corner point might add more (new) fractional vertices.

In higher dimensions, it could add many more new nonintegral vertices!



But it could happen that we do not add any new non-integral vertices by adding the cut. Indeed, such cuts are the tightest ones you could add.



It may not be possible to always add a tightest cut — we still benefit from cutting off fractional corner points, so we study cutting planes in general...

<u>Recall</u>: $\bar{a}^T\bar{x} \leq \beta$ is valid for $P \subseteq \mathbb{R}^n$ if $\bar{a}^T\bar{x} \leq \beta \ \forall \bar{x} \in P$.

# Chvátal-Gromory (CG) Cuts

Vásek Chvátal ("Vashek HoTal").

Most other classes of cuts could be derived by applying the CG cut procedure repeatedly.

## Pure integer case

(1) $Y = \{\bar{x} \in \mathbb{Z}^n \mid A\bar{x} \le \bar{b}\}$.

$\bar{u} \geq \bar{0} \Rightarrow (\bar{u}^T A)\bar{x} \le \bar{u}^T \bar{b}$ is valid for $Y$.

If $\bar{u}^T A$ is integral, then

$$(\bar{u}^T A)\bar{x} \le \lfloor \bar{u}^T \bar{b} \rfloor \text{ is valid for } Y.$$

(2) $P = \{\bar{x} \in \mathbb{R}^n \mid A\bar{x} \le \bar{b}, \bar{x} \geq \bar{0}\}$.

$\bar{u} \geq \bar{0} \Rightarrow \underbrace{(\bar{u}^T A)\bar{x} \le \bar{u}^T \bar{b}, \text{ is valid for } P.}_{(1)}$

$\Rightarrow \underbrace{\lfloor(\bar{u}^T A)\rfloor \bar{x} \le \bar{u}^T \bar{b}, \text{ is valid for } P}_{(2)}$

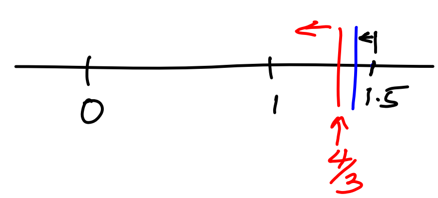
Since $\bar{x} \geq \bar{0}$, (2) weakens (1).

Hence $\lfloor \bar{u}^T A \rfloor \bar{x} \le \lfloor \bar{u}^T \bar{b} \rfloor$ is valid for $Y = P \cap \mathbb{Z}^n$.

Example $\quad \lfloor 3.3 \rfloor x \le 4.5$ is valid for $P$

$\Rightarrow \quad 3x \le \lfloor 4.5 \rfloor$ is valid for $P$

$\quad 3x \le 4$ is valid for $P \cap \mathbb{Z}$.



# Mixed Integer Case (of the CG cut)

## Mixed integer rounding (MIR)

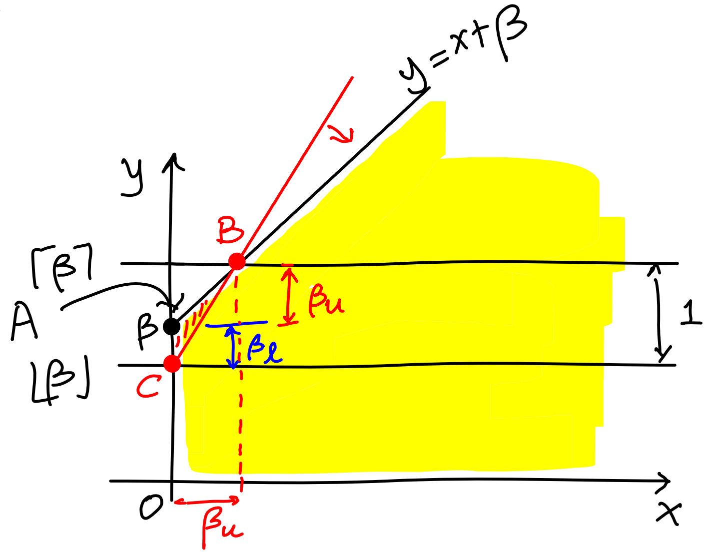
$X = \{(x,y) \in \mathbb{R}_{\geq 0} \times \mathbb{Z} \mid y \leq x + \beta\}$.

interesting case: $\beta$ is non-integral.

$X = \{(x,y) \in \mathbb{R}_{\geq 0} \times \mathbb{Z} \mid y \leq x+\beta\}$

$\beta$ is non-integral.

$A(0,\beta)$ needs to be cut off.




### Notation

$\beta_\ell = \beta - \lfloor\beta\rfloor$
$\beta_u = \lceil\beta\rceil - \beta$

} lower and upper fractional parts.

e.g., $\beta = 13.3$,
$\beta_\ell = 0.3$, $\beta_u = 0.7$.

At B, $y = \lceil\beta\rceil = \beta + \beta_u$. With $y = x+\beta$, we get

$\beta + \beta_u = x + \beta \Rightarrow x = \beta_u$.

The cut is $y = mx + \lfloor\beta\rfloor$, and at $B(\beta_u, \lceil\beta\rceil)$, we get

$$\lceil\beta\rceil = \lfloor\beta\rfloor + 1 = m\beta_u + \lfloor\beta\rfloor$$

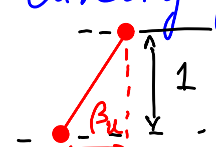
$$\Rightarrow m = \frac{1}{\beta_u}.$$

Hence we get that

$$y \leq \frac{1}{\beta_u} x + \lfloor\beta\rfloor$$

is valid for $X$.

Alternatively, $m = \frac{1}{\beta_u}$ directly from the figure




$x \geq 0$ is needed to get the fractional corner point $A(0,\beta)$ in the first place.

# MATH 567 : Lecture 17 (03/06/2025)

Today :
* MIG cuts
* Knapsack cuts
* cover inequalities

---

## Mixed-integer Gomory Cut (MIG cut)

Extension of MIR to higher dimensions. Let

$$X = \left\{ (\bar{x}, \bar{y}) \in \mathbb{R}^m_{\geq 0} \times \mathbb{Z}^n_{\geq 0} \;\middle|\; \sum_{j \in N} a_j y_j + \sum_{j \in M} a_j x_j = \beta \right\}$$

where $N = \{1, 2, \dots, n\}$ and $M = \{1, 2, \dots, m\} + n$.

$\longrightarrow$ $N, M$ are indexs sets for $\bar{y}, \bar{x}$, resp.

IDEA: Derive a valid inequality of the form $y \leq x + \beta$ from the equation defining $X$, and apply MIR.

$$\sum_{j \in N} a_j y_j + \sum_{j \in M} a_j x_j = \beta$$

we ignore terms with $a_j \geq 0$ ($x_j \geq 0 \; \forall j$)

$$\Rightarrow \sum_{j : a_{j_\ell} \leq \beta_\ell} \lfloor a_j \rfloor y_j + \sum_{j : a_{j_\ell} > \beta_\ell} \underbrace{a_j}_{\lceil a_j \rceil - a_{j_u}} y_j + \sum_{a_j < 0} a_j x_j \leq \beta$$

$$\Rightarrow \underbrace{\left( \sum_{j : a_{j_\ell} \leq \beta_\ell} \lfloor a_j \rfloor y_j + \sum_{j : a_{j_\ell} > \beta_\ell} \lceil a_j \rceil y_j \right)}_{y} \leq \beta + \underbrace{\left( \sum_{j : a_{j_\ell} > \beta_\ell} a_{j_u} y_j - \sum_{a_j < 0} a_j x_j \right)}_{x}$$

We now apply MIR to get $y \leq \frac{1}{\beta_u} x + \lfloor \beta \rfloor$.

$$\Rightarrow \left(\sum'_{a_{j_\ell}\le\beta_\ell}\lfloor a_j\rfloor y_j + \sum'_{a_{j_\ell}>\beta_\ell}\underbrace{\lceil a_j\rceil}_{\lfloor a_j\rfloor+1} y_j\right) \le \lfloor\beta\rfloor + \left(\sum'_{a_{j_\ell}>\beta_\ell}\frac{a_{ju}}{\beta_u}y_j - \sum'_{a_j<0}\frac{a_j}{\beta_u}x_j\right)$$

$$\Rightarrow \sum'_{a_{j_\ell}\le\beta_\ell}\lfloor a_j\rfloor y_j + \sum'_{a_{j_\ell}>\beta_\ell}\left(\lfloor a_j\rfloor + \left(\frac{\beta_u - a_{ju}}{\beta_u}\right)\right)y_j + \frac{1}{\beta_u}\sum'_{a_j<0}a_j x_j \le \lfloor\beta\rfloor$$

is the MIG cut.

<u>Note</u>: If $|M|=0$, i.e., there are no $x_j$'s, the usual CG cut

$$\sum'_{j\in N}\lfloor a_j\rfloor y_j \le \lfloor\beta\rfloor.$$

But the MIG cut gives

$$\sum'_{a_{j_\ell}\le\beta_\ell}\lfloor a_j\rfloor y_j + \sum'_{a_{j_\ell}>\beta_\ell}\left(\lfloor a_j\rfloor + \underbrace{\left(\frac{\beta_u - a_{ju}}{\beta_u}\right)}_{\ge 0}\right)y_j \le \lfloor\beta\rfloor,$$

which is stronger than the CG cut.

Wolsey (Integer Programming) calls the MIR cut as the "basic mixed integer inequality", and the special case of MIG cut with $|M|=1$, $|N|=2$ as the "MIR inequality".

## Example

$$X = \left\{ (\bar{x},\bar{y}) \in \mathbb{R}^2_{\geq 0} \times \mathbb{Z}^3_{\geq 0} \;\middle|\; \underbrace{2}_{a_4}x_1 \underbrace{-}_{a_5}x_2 + \underbrace{\tfrac{10}{3}}_{a_1}y_1 + \underbrace{\phantom{1}}_{a_2}y_2 + \underbrace{\tfrac{11}{4}}_{a_3}y_3 = \underbrace{\tfrac{21}{2}}_{\beta} \right\}.$$

$$a_{1_l} = \frac{1}{3},\ a_{1_u} = \frac{2}{3},\ a_{3_l} = \frac{3}{4},\ a_{3_u} = \frac{1}{4},\ \beta_l = \beta_u = \frac{1}{2}.$$

$$\Rightarrow \frac{10}{3}y_1 + y_2 + \frac{11}{4}y_4 - x_2 \leq \frac{21}{2} \quad \text{is valid for } X.$$

Note that we have removed the $2x_1$ term from lhs.

$$\Rightarrow \left\lfloor \frac{10}{3} \right\rfloor y_1 + y_2 + \left( \left\lfloor \frac{11}{4} \right\rfloor + \frac{\left(\frac{1}{2} - \frac{1}{4}\right)}{1/2} \right) y_3 - \frac{1}{1/2}x_2 \leq \left\lfloor \frac{21}{2} \right\rfloor$$

is valid for $X$.

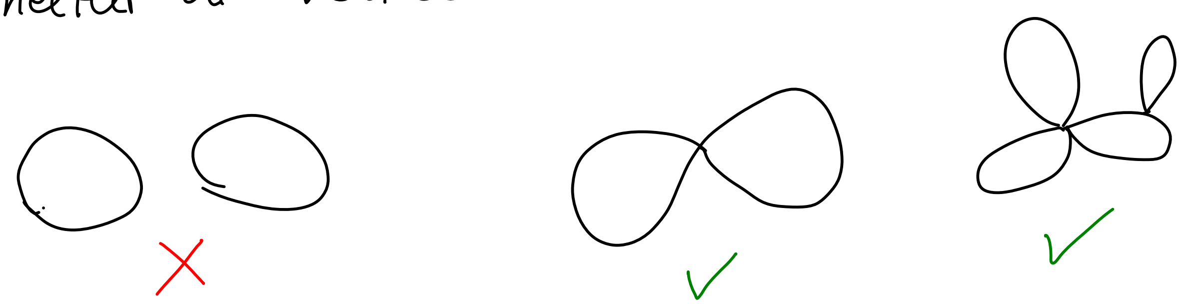
$$\Rightarrow 3y_1 + y_2 + \frac{5}{2}y_3 - 2x_2 \leq 10 \quad \text{is valid for } X.$$

# Project 1: Hiker's tour problem (HTP)

$G = (V, E)$ directed graph

Find a circuit (closed walk) with following properties:

* start and end at a given vertex;
* do not have to visit every $v \in V$;
* could visit a node more than once;
* subtours are allowed as long as they are connected at vertices.



* $\sum_{(i,j) \in W} c_{ij} \geq L$ ← data

Come up with formulations similar to the MTZ and subtour formulations for TSP.

# Knapsack Cuts for pure 0-1 Programs

IDEA:

Given $\left\{ \begin{array}{ll} \max & \bar{c}^T\bar{x} \\ \text{s.t.} & A\bar{x} \le \bar{b} \\ & \bar{x} \in \{0,1\}^n \end{array} \right\}$, pick $\bar{a}^T\bar{x} \le \beta$ from $A\bar{x} \le \bar{b}$,

generate cuts for $Y = \{\bar{x} \in \{0,1\}^n \mid \bar{a}^T\bar{x} \le \beta\}$, and add these cuts to the original IP.

Assume $a_i, \beta \in \mathbb{Z}$. WLOG, assume $a_i \ge 0 \ \forall i$ (in $\bar{a}$). If $a_1 < 0$, we could replace $x_1$ with $(1-x_1)$ and $a_1$ with $-a_1$ to get another inequality, for instance.

We define covers that capture the subsets of $a_i$ that add to values larger than $\beta$ (and hence "cover" it). If their sum is $> \beta$, we cannot have all the corresponding $x_j = 1$, which is the cut we are seeking.

**Def** $C \subseteq \{1, 2, \ldots, n\} = N$ is a **cover** if $\bar{a}(C) > \beta$, where $\bar{a}(C) = \sum_{i \in C} a_i$. Further, we say that $C$ is a **minimal cover** if $C$ is a cover, but $C \setminus \{i\}$ is not a cover $\forall i \in C$.

## Example

Let $Y = \{\bar{x} \in \{0,1\}^7 \mid 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$ $\circledast$

$C_1 = \{1,4,5\}$ is a minimal cover.

$C_2 = \{3,4,5,6\}$ is a minimal cover.

$C_3 = \{3,4,5,6,7\}$ is a cover, but is not minimal.

Note that $a_3 + a_4 + a_5 + a_6 + a_7 = 21 > \beta = 19$. But so is $a_3 + a_4 + a_5 + a_6 (= 20)$.

**Claim** $C$ is a cover $\Longrightarrow$ $\bar{x}(C) \leq |C| - 1$ is valid for $Y$.

Here, $\bar{x}(C) = \sum_{j \in C} x_j$.

$C_1$: $x_1 + x_4 + x_5 \leq 2$ is valid for $Y$. — (1)

$C_2$: $x_3 + x_4 + x_5 + x_6 \leq 3$ is valid for $Y$ — (2)

$C_3$: $x_3 + x_4 + x_5 + x_6 + x_7 \leq 4$ is valid for $Y$, — (3).

But (3) is weaker than (2), e.g., $x_3 + x_4 + x_5 + x_6 = 3.5$ satisfies (3), but violates (2).

Notice we added an extra variable ($x_7$) to the lhs of the $\leq$ inequality with $x_7 \geq 0$, but also increased the rhs by 1. If we could add more nonnegative terms to the lhs while not changing the rhs, then we will strengthen the cut.

**Def** The **extension** of a cover $C$ is

$$E(C) = \{ j \notin C, j \in N \mid a_j \geq \max_{i \in C} \{a_i\} \} \cup C.$$

e.g., $E(\{3,4,5,6\}) = \{1,2,3,4,5,6\}$.

**Claim** $\bar{x}(E(C)) \leq |C| - 1$ is valid for $Y$.

So, $x_3 + x_4 + x_5 + x_6 \leq 3$ — (2) can be strengthened

to $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$ — (4).

Since $\bar{a}(C) > \beta$, and the added to $C$ to obtain $E(C)$ are such that $a_j \geq \max_{i \in C}(a_i)$, the validity of the new cut follows.

# MATH 567: Lecture 18 (03/18/2025)

Today: * lifted cover inequalities
* separation problem

Recall definitions on knapsack cover inequalities:

**Def** $C \subseteq \{1,2,\dots,n\} = N$ is a **cover** if $\bar{a}(C) > \beta$, where $\bar{a}(C) = \sum_{i \in C} a_i$. Further, we say that $C$ is a **minimal cover** if $C$ is a cover, but $C \setminus \{i\}$ is not a cover $\forall i \in C$.

Let $Y = \{\bar{x} \in \{0,1\}^7 \mid 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$ $(*)$

$C_1 = \{1,4,5\}$ is a minimal cover.

$C_2 = \{3,4,5,6\}$ is a minimal cover.

$C_3 = \{3,4,5,6,7\}$ is a cover, but is not minimal.

**Claim** $C$ is a cover $\Rightarrow$ $\bar{x}(C) \leq |C|-1$ is valid for $Y$.

Here, $\bar{x}(C) = \sum_{j \in C} x_j$.

$C_1$: $x_1 + x_4 + x_5 \leq 2$ is valid for $Y$. (1)

$C_2$: $x_3 + x_4 + x_5 + x_6 \leq 3$ is valid for $Y$ (2)

$C_3$: $x_3 + x_4 + x_5 + x_6 + x_7 \leq 4$ is valid for $Y$, (3)

**Def** The **extension** of a cover $C$ is

$$E(C) = \{j \notin C, j \in N \mid a_j \geq \max_{i \in C} \{a_i\}\} \cup C.$$

e.g., $E(\{3,4,5,6\}) = \{1,2,3,4,5,6\}$.

**Claim** $\bar{x}(E(C)) \leq |C| - 1$ is valid for $Y$.

So, $x_3+x_4+x_5+x_6 \leq 3$ —— (2) can be strengthened to

$$x_1+x_2+x_3+x_4+x_5+x_6 \leq 3 \quad \text{———— (4).}$$

This is an extended cover cut/inequality.

But, $2x_1+x_2+x_3+x_4+x_5+x_6 \leq 3$, is also valid for $Y$. —— (5)

Recall, $Y = \{\bar{x} \in \{0,1\}^7 \mid 11x_1+6x_2+6x_3+5x_4+5x_5+4x_6+x_7 \leq 19\}$ (*)

(5) holds, as $x_1 = 1 \Rightarrow (x_2+\cdots+x_6) \leq 1$ $(19-11=8)$.

Note that (5) is stronger than (4).

How did we get (5)? By **lifting** coefficient(s)!

We lifted the coefficient of $x_1$ from 1 to 2.

In a more general setting, we could lift the coefficient of some $x_j$ from 0 to the largest possible value. Also, the idea of lifting could be applied to other classes of inequalities as well, and not just for covers.

Given a cover $C_1$ with $1 \notin C_1$, we know $\bar{x}(C) \leq |C|-1$ is a valid inequality for $(\bar{a}^T\bar{x})(C_1) \leq \beta$, where $(\bar{a}^T\bar{x})(C_1) = \sum_{j \in C_1} a_j x_j$. We want $\alpha_1$ such that

$\alpha_1 x_1 + \bar{x}(C_1) \leq |C|-1$ is valid for $a_1x_1 + (\bar{a}^T\bar{x})(C_1) \leq \beta$.

If $x_1 = 0$, $\alpha_1$ can be any valid value ($\alpha_1 \geq 0$).

If $x_1 = 1$, $\alpha_1 + \bar{x}(C) \leq |C| - 1$ should hold for all $\bar{x} \in \{0,1\}^n$ such that $a_1 + (\bar{a}^T \bar{x})(C_1) \leq \beta$.

let $z = \left\{ \begin{array}{ll} \max & \bar{x}(C_1) \\ \text{s.t.} & (\bar{a}^T \bar{x})(C_1) \leq \beta - a_1 \\ & \bar{x} \in \{0,1\}^n \end{array} \right\}$ — (KP)

Then we have $z \leq |C_1| - 1 - \alpha_1 \Rightarrow \boldsymbol{\alpha_1 \leq |C| - 1 - z}$, an upper bound on $\alpha_1$. The best $\alpha_1$ is $|C| - 1 - z$, but by choosing $z = z_u$, the LP-relaxation objective function value of (KP), we still get a good value for $\alpha_1$.

In general, we do not want to solve a subproblem as an IP — always solve only LPs as subproblems.

So, we set $\alpha_1 = |C| - 1 - z_u$.

## Example

$$Y = \{\bar{x} \in \{0,1\}^7 \mid 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\} \quad (*)$$

Consider $C_2 = \{3,4,5,6\} \not\ni 1$.

(KP) here is $z = \left\{ \begin{array}{ll} \max & x_3 + x_4 + x_5 + x_6 \\ \text{s.t.} & 6x_3 + 5x_4 + 5x_5 + 4x_6 \leq 19 - 11 = 8 \\ & x_3, x_4, x_5, x_6 \in \{0,1\} \end{array} \right\}$.

$z = 1$ here ($x_j = 1$ for any one $j \in C_2$). $\Rightarrow \alpha_1 = |C_2| - 1 - z$
$= 4 - 1 - 1 = 2$.

Solving the LP relaxation of (KP), we get

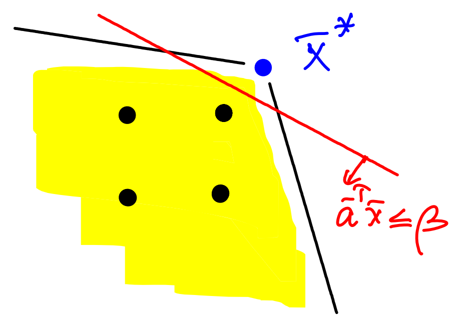
$z_u = 1.8$ ($x_6 = 1$, and $x_5 = 0.8$ or $x_4 = 0.8$) $\Rightarrow \alpha_1 = |C_2| - 1 - z_u = 1.2$, (which is still better than 1). So, the new lifted cover inequality is $1.2x_1 + x_3 + x_4 + x_5 + x_6 \leq 3$.

How did I get $z_u = 1.8$? Essentially using a "greedy" approach to solve the knapsack problem.

$$\begin{aligned} \max \quad & c_1x_1 + \cdots + c_nx_n \qquad c_j, a_j \geq 0 \\ \text{s.t.} \quad & a_1x_1 + \cdots + a_nx_n \leq \beta \\ & 0 \leq x_j \leq u_j \end{aligned}$$

Sort the $x_j$'s in the decreasing order of $\frac{c_j}{a_j}$, and set $x_j$'s to $\min\{u_j, \beta'/a_j\}$, where $\beta'$ is the "updated" $\beta$, i.e, $\beta \leftarrow \beta - a_i x_i$ after setting $x_i$ in the previous step.

## Separation Problem




In general, for any combinatorial optimization problem (COP): $\max = \{\bar{c}^T\bar{x} \mid \bar{x} \in X \subseteq \mathbb{R}^n\}$,

and given $\bar{x}^* \in \mathbb{R}^n$, is $\bar{x}^* \in \text{conv}(X)$? If YES, prove it. If NO, find an inequality $\bar{a}^T\bar{x} \leq \beta$ satisfied by all $\bar{x} \in X$, but is violated by $\bar{x}^*$, i.e, $\bar{a}^T\bar{x}^* > \beta$.

The inequality with $(\bar{a}^T\bar{x}^* - \beta)$ largest is the "most violated" separating inequality.

We consider the separation problem in the context of knapsack cover inequalities.

Let $Y = \{\bar{x} \in \{0,1\}^n \mid \bar{a}^T\bar{x} \le \beta\}$, $a_i, \beta \in \mathbb{Z}_{\geq 0}$, and let $\bar{x}^* \in \mathbb{R}^n$, but $\bar{x}^* \notin \{0,1\}^n$, i.e., $0 < x_j^* < 1$ for at least one $j \in N$. We want to separate $\bar{x}^*$ using a cover inequality, i.e., find a cover $C$ such that $(\bar{a}^T\bar{x}^*)(C) > \beta$ and $\bar{x}^*(C) > |C| - 1$.

Define $\bar{y} \in \{0,1\}^n$ as the incidence vector of $C$. We need

$$\left\{ \begin{array}{c} \sum_{j=1}^n x_j^* y_j > \sum_{j=1}^n y_j - 1 \\ \sum_{j=1}^n a_j y_j > \beta \\ y_j \in \{0,1\} \ \forall j \end{array} \right\}$$

$$\Longleftrightarrow \left\{ \begin{array}{c} 1 > \sum_{j=1}^n (1 - x_j^*) y_j \\ \sum_{j=1}^n a_j y_j \geq \beta + 1 \\ y_j \in \{0,1\} \ \forall j \end{array} \right\}$$

as $a_j, \beta \in \mathbb{Z}_{\geq 0}$

So, we can find

$$z = \left\{ \begin{array}{ll} \min & \sum_{j=1}^n (1 - x_j^*) y_j \\ \text{s.t.} & \sum_{j=1}^n a_j y_j \geq \beta + 1 \\ & y_j \in \{0,1\} \ \forall j \end{array} \right\}.$$

If $z<1$, the cover we seek exists, and its incidence vector is given by $\bar{y}$. Hence $\bar{x}^*$ violates the cover inequality $\bar{x}(C) \leq |C|-1$.

Example

$$Y = \{\bar{x} \in \{0,1\}^7 \mid 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\} \quad (*)$$

Let $\bar{x}^* = [\overset{1}{0}, \overset{2}{0}, \overset{3}{1}, \overset{4}{1}, \overset{5}{1}, \overset{6}{\tfrac{3}{4}}, \overset{7}{0}]^T$. Find a separating cover for $\bar{x}^*$.

We solve

$$\min \quad z = y_1 + y_2 + \tfrac{1}{4} y_6 + y_7$$
$$\text{s.t.} \quad 11y_1 + 6y_2 + 6y_3 + 5y_4 + 5y_5 + 4y_6 + y_7 \geq 20$$
$$y_j \in \{0,1\}, \ j = 1, \dots, 7.$$

Optimal solution: $\bar{y} = [0,0,1,1,1,1,0]$, $z^* = \frac{1}{4}$.

Hence $\bar{x}^*$ violates $x_3 + x_4 + x_5 + x_6 \leq 3$.

Indeed, $\bar{x}^*(C) = 3\frac{3}{4} \not\leq 3$.

Note that a greedy approach gives the optimal integer solution for this knapsack problem!

# MATH 567: Lecture 19 (03/20/2025)

Today: * disjunctive cuts

## Disjunctive Cuts (for 0-1 IPs)

IDEA: Derive cuts by first creating a non-linear system then linearizing the same by going to higher dimensions, and then projecting back.

If $x_j \in \{0,1\} \; \forall j$, $\quad x_j^2 \longleftarrow x_j$, $\quad x_i x_j \longleftarrow y_{ij} \in \{0,1\}$.

Let $P_i = \{\bar{x} \mid A_i \bar{x} \le \bar{b}^i\}$, $i = 1,2$, $P_i \neq \emptyset$.

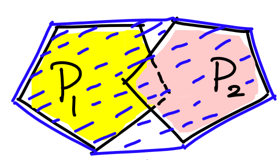
Assume $\text{rec}(P_1) = \text{rec}(P_2)$. Thus the sharp representation for $P_1 \cup P_2$ is

$$\left.\begin{aligned} A_1 \bar{x}^1 &\le \bar{b}^1 y_1 \\ A_2 \bar{x}^2 &\le \bar{b}^2 y_2 \\ \bar{x} &= \bar{x}^1 + \bar{x}^2 \\ y_1 + y_2 &= 1 \\ y_1, y_2 &\in \{0,1\} \end{aligned}\right\} \text{——— } *\text{-sharp}$$

Recall ① $\bar{x} \in P_1 \cup P_2 \iff \exists\, (\bar{x}^1, \bar{x}^2, y_1, y_2)$ such that $(\bar{x}, \bar{x}^1, \bar{x}^2, y_1, y_2)$ satisfies $*$-sharp.

② $\text{Proj}_{\bar{x}}$ (LP-relaxation of $*$-sharp) $= \text{conv}(P_1 \cup P_2)$.

if you project out $\bar{x}^1, \bar{x}^2, y_1, y_2$




$\text{conv}(P_1 \cup P_2)$

Idea: we create a non linear system from the original system, then linearize by adding more variables, and finally project back to the original space to derive valid inequalities.

# Lovász-Schrijver (LS) Procedure (for 0-1 IPs)

$$X = \{\bar{x} \in \mathbb{Z}^n \mid A\bar{x} \leq \bar{b}\}$$ includes $0 \leq x_j \leq 1$

$$K = \{\bar{x} \in \mathbb{R}^n \mid A\bar{x} \leq \bar{b}\}$$

1. Select $j \in \{1, 2, \ldots, n\}$.

2. Create the non-linear system

$$\left.\begin{aligned} (A\bar{x} - \bar{b})\, x_j &\leq \bar{0} \\ (A\bar{x} - \bar{b})(1 - x_j) &\leq \bar{0} \\ x_j(1 - x_j) &= 0 \end{aligned}\right\} M_j^{NL}(K)$$

holds, as $x_j \in \{0, 1\}$

non linear, as there are quadratic terms $x_i x_j$

3. Linearize the system by replacing $x_j^2$ by $x_j$, and $x_i x_j$ for $j \neq i$ by $y_i$ (where $y_i$ is supposed to be binary). The polyhedron thus obtained is $M_j(K)$.

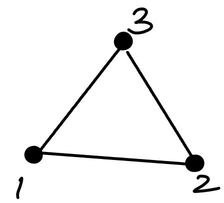
4. Let $P_j(K) = \text{Proj}_{\bar{x}}(M_j(K))$. has the cut(s) we seek

The LS and other similar procedures have many theoretical and computational applications. A standard question is whether we could get the required cut by a small (i.e., polynomial) number of applications (repeatedly) of the LS procedure.

<u>Example</u> Vertex packing problem (also called the maximal independent set problem – select the largest subset of vertices so that no two of the vertices are joined by an edge).

e.g.,

$$X \left\{ \begin{array}{l} \left.\begin{array}{l} x_1+x_2 \le 1 \\ x_2+x_3 \le 1 \\ x_1+x_3 \le 1 \\ 0 \le x_i \le 1,\ i=1,2,3 \end{array}\right\} K \\ \quad x_i \in \mathbb{Z} \end{array}\right.$$




Want to derive
$x_1+x_2+x_3 \le 1$

Apply LS procedure with $j=1$:

$$x_1^2 + x_1x_2 \le x_1 \quad , \quad \text{replace } x_1^2 \text{ by } x_1$$

$$\Rightarrow \ x_1x_2 \le 0$$

But from $-x_2 \le 0$, we get $-x_1x_2 \le 0 \Rightarrow x_1x_2 \ge 0$.

$$\Rightarrow x_1x_2 = 0.$$

Similarly, $x_1x_3 = 0$.

Consider $(x_2+x_3 \le 1)(1-x_1)$:

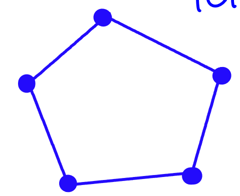
$$x_2+x_3 - \underbrace{\cancel{x_1x_2}}_{=0} - \underbrace{\cancel{x_1x_3}}_{=0} \le 1-x_1$$

$$\Rightarrow \quad \boxed{x_1+x_2+x_3 \le 1}$$

This is an instance of "odd-hole" inequality.

**Def** An **odd hole** is $C \subseteq V$ with $|C|$ odd, with edges connecting the vertices making a simple cycle, i.e., a "hole".

$|C|=5$



We can pick at most $\frac{|C|-1}{2}$ nodes, i.e.,

$$\bar{x}(C) \leq \frac{|C|-1}{2}$$ is valid, and is

derivable by the LS procedure using $M_j(C)$ for any $j \in C$.

$\sum_{(i,j)\in C} x_{ij} \leq 2$ here.

We could also derive this inequality by adding $x_i + x_j \leq 1$ over $C$, which gives

$$2\,\bar{x}(C) \leq |C|,$$ which we can

divided by 2, and round down (CG procedure) to get

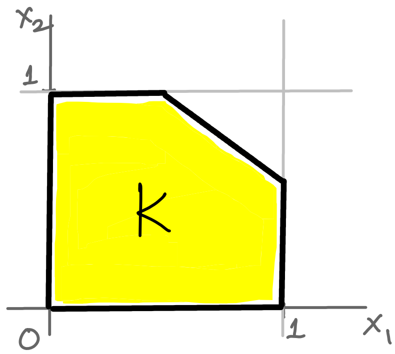
$$\bar{x}(C) \leq \left\lfloor \frac{|C|}{2} \right\rfloor = \frac{|C|-1}{2}.$$

But there are other problem instances where the LS procedure gives inequalities which cannot be derived by other procedures.

**Theorem 14** If $Q_j(K) = \text{conv}\left([K \cap (x_j=0)] \cup [K \cap (x_j = 1)]\right)$ then $P_j(K) = Q_j(K)$.

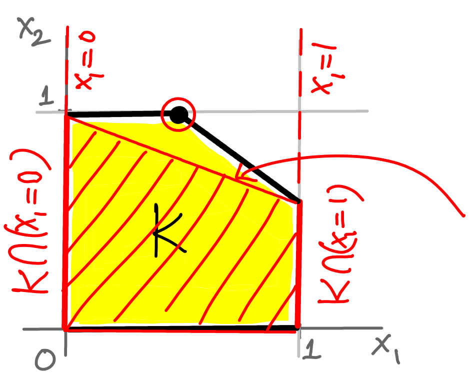
Theorem 13 was on reduced cost fixing in lecture 15!

Before presenting the proof, we illustrate the concept in 2D. Consider a non-trivial polytope in the unit square.




Consider $Q_1(K) = \text{conv}\left([K \cap (x_1=0)] \cup [K \cap (x_1=1)]\right)$.

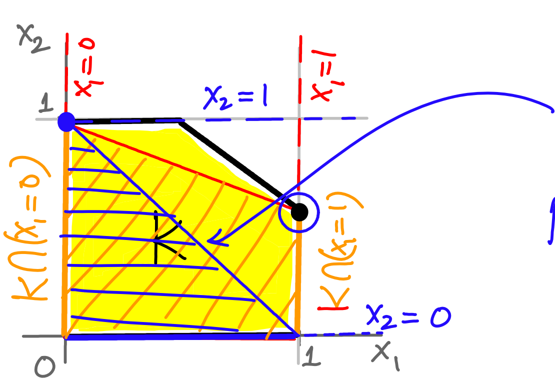
Note: the polytope need not be "symmetric" — or show the same behavior




$Q_1(K) = \text{conv}\left([K \cap (x_1=0)] \cup [K \cap (x_1=1)]\right)$

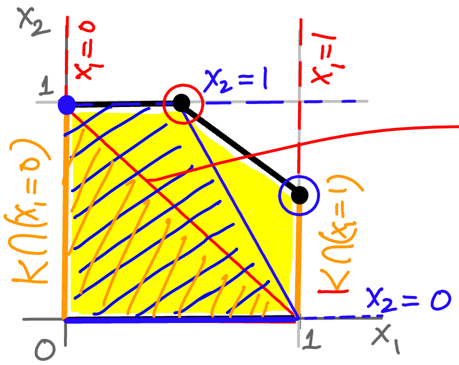
Notice how the fractional point ⊙ is cut off.

We could now apply the same procedure again using $j=2$ to get the tightest polytope. In detail, we consider $Q_2(Q_1(K))$.




$Q_2(Q_1(K))$

Notice that the other fractional corner point ⊙ is also cutoff now.




$Q_1(Q_2(K)) = Q_2(Q_1(K))$

the order does not matter!

Recall Theorem 14: $P_j(K) = Q_j(K) := \text{conv}\left([K\cap(x_j=0)] \cup [K\cap(x_j=1)]\right)$.

Proof ($\Leftarrow$) $\quad P_j(K) \supseteq Q_j(K)$

We try to show $K\cap(x_j=0) \subseteq P_j(K)$

and $K\cap(x_j=1) \subseteq P_j(K)$.

If $\bar{x}' \in K$ and $x_j'$ is either 0 or 1, then

$$\left.\begin{array}{l} A\bar{x}' \leq \bar{b} \\ x_j' \geq 0,\ 1-x_j' \geq 0 \\ x_j'(1-x_j') = 0 \end{array}\right\} \text{ all hold.}$$

So, we can indeed form the system $M_j^{N2}(K)$

$$\left\{\begin{array}{l} (A\bar{x}-\bar{b})x_j' \leq \bar{0} \\ (A\bar{x}-\bar{b})(1-x_j') \leq \bar{0} \\ x_j'(1-x_j') = 0 \end{array}\right\},$$

eliminate nonlinear terms, linearize, and project to get $P_j(K)$.

$$\Rightarrow \quad Q_j(K) \subseteq P_j(K).$$

$(\Rightarrow)\quad P_j(K) \subseteq Q_j(K).$

We show that $P_j(K)$ contains the sharp formulation of union of polyhedra, whose convex hull is $Q_j(K)$.

$$M_j(K) \text{ has } \left\{ \begin{array}{c} (A\bar{x}-\bar{b})x_j \le \bar{0} \\ (A\bar{x}-\bar{b})(1-x_j) \le \bar{0} \\ x_j(1-x_j) = 0 \end{array} \right\} \qquad \begin{array}{c} A\overbrace{\bar{x}x_j}^{\bar{x}^1} - \bar{b}\overbrace{x_j}^{y_1} \le \bar{0} \\ A\overbrace{\bar{x}(1-x_j)}^{\bar{x}^2} - b\underbrace{(1-x_j)}_{y_2} \le \bar{0} \\ [\bar{x}(1-x_j)]_j = 0 \end{array}$$

Write $\bar{x}x_j$ as $\bar{x}^1$, $\bar{x}(1-x_j)$ as $\bar{x}^2$, $x_j \leftarrow y_1$, $(1-x_j) \leftarrow y_2$

$$\Rightarrow \quad A\bar{x}^1 \le \bar{b}y_1$$
$$A\bar{x}^2 \le \bar{b}y_2$$

$$\bar{x} - \bar{x}x_j - \bar{x}(1-x_j) = \bar{0} \Rightarrow \bar{x} = \bar{x}^1 + \bar{x}^2$$

$$x_j + (1-x_j) = 1 \Rightarrow y_1 + y_2 = 1$$

$j^{th}$ unit vector

$$x_j^2 - x_j = 0 \Rightarrow (\bar{x}^1)_j = y_1 \equiv \bar{e}_j^T \bar{x}^1 = y_1$$

$$[\bar{x}(1-x_j)]_j = 0 \Rightarrow \bar{x}^2_j = 0 \equiv \bar{e}_j^T\bar{x}^2 = 0 = 0\cdot y_2$$

$$\boxed{\begin{array}{c} A\bar{x}^1 \le \bar{b}y_1 \\ \bar{e}_j^T\bar{x}^1 = y_1 \\ A\bar{x}^2 \le \bar{b}y_2 \\ \bar{e}_j^T\bar{x}^2 = 0\cdot y_2 \\ \bar{x} = \bar{x}^1 + \bar{x}^2 \\ y_1 + y_2 = 1 \end{array}}$$

polyhedron of the sharp formulation of $P_1 \cup P_2$

where $P_1 = \{\bar{x} \mid A\bar{x} \le \bar{b},\ \underbrace{\bar{e}_j^T\bar{x} = 1}_{x_j = 1}\}$ and $P_2 = \{\bar{x} \mid A\bar{x} \le \bar{b},\ \underbrace{\bar{e}_j^T\bar{x} = 0}_{x_j = 0}\}$.

$$\Rightarrow \quad P_j(K) \subseteq Q_j(K).$$

$\square$

# MATH 567: Lecture 20 (03/25/2025)

Today: * A different proof for $P_j(K) \subseteq Q_j(K)$
* Disjunctive programming

**Recall** $Q_j(K) = \text{conv}([K \cap (x_j=0)] \cup [K \cap (x_j=1)])$; **Theorem 14** $P_j(K) = Q_j(K)$.

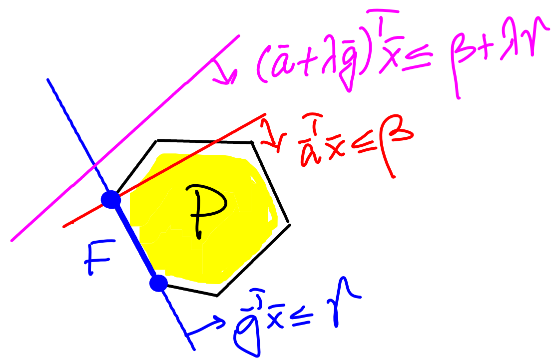
**A different proof** We first state and prove a lemma.

**Lemma 15** Let $P = \{\bar{x} \in \mathbb{R}^n \mid A\bar{x} \le \bar{b}, \bar{g}^T\bar{x} \le \gamma\}$ and
$F = P \cap \{\bar{x} \in \mathbb{R}^n \mid \bar{g}^T\bar{x} = \gamma\}$, i.e.,
F is a face of P. Suppose $\bar{a}^T\bar{x} \le \beta$ is valid for F but not valid for P. Then there exists $\lambda \ge 0$ such that
$(\bar{a} + \lambda\bar{g})^T\bar{x} \le \beta + \lambda\gamma$ is valid for P.

**Proof** (Farkas' lemma)

$$F \left\{ \begin{array}{ll} A\bar{x} \le \bar{b} & \bar{u} \ge \bar{0} \\ \bar{g}^T\bar{x} \le \gamma & v_1 \ge 0 \\ -\bar{g}^T\bar{x} \le -\gamma & v_2 \ge 0 \end{array} \right.$$

(the first two inequalities define $P$)




Can derive $\bar{a}^T\bar{x} \le \beta$ from F:

$$\Rightarrow \left.\begin{array}{l} \bar{u}^TA + v_1\bar{g}^T - v_2\bar{g}^T = \bar{a}^T \\ \bar{u}^T\bar{b} + v_1\gamma - v_2\gamma \le \beta \end{array}\right\} \quad \begin{array}{l} \bar{u}^TA + v_1\bar{g}^T = \bar{a}^T + v_2\bar{g}^T \\ \bar{u}^T\bar{b} + v_1\gamma \le \beta + v_2\gamma \end{array}$$

Another inequality can be derived using multipliers $(\bar{u}, v_1)$ from P:

$\Rightarrow (\bar{a} + v_2\bar{g})^T\bar{x} \le \beta + v_2\gamma$ is valid for P.

$\lambda = v_2$ works for the lemma. $\square$

## Proof for $P_j(K) \subseteq Q_j(K)$

Suppose $\bar{a}^T\bar{x} \le \beta$ is valid for both $K \cap (x_j = 0)$ and $K \cap (x_j = 1)$, two faces of $K$.

$-x_j \le 0$ $\qquad$ $x_j \le 1$ or $-(1-x_j) \le 0$

We use Lemma 15 to simultaneously lift this inequality so that it is valid for all of $K$.

$\Rightarrow$ Find $\lambda \geq 0$ and $\mu \geq 0$ such that

$$\bar{a}^T\bar{x} - \lambda x_j \le \beta \text{ is valid for } K, \qquad (1)$$

and $$\bar{a}^T\bar{x} - \mu(1-x_j) \le \beta \text{ is valid for } K. \qquad (2)$$

---

WLOG, (1) and (2) are already part of $A\bar{x} \le \bar{b}$. Else, we could derive them from $A\bar{x} \le \bar{b}$ using nonnegative multipliers.

Consider the following scaled inequalities in $M_j^{NL}(K)$

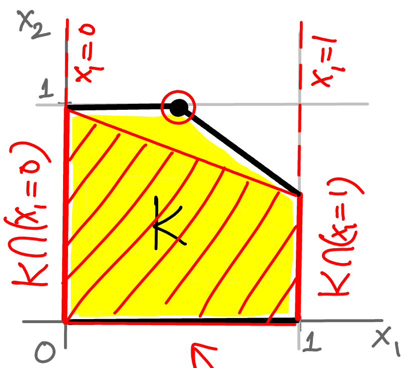
$$\left.\begin{cases} (1-x_j)(\bar{a}^T\bar{x} - \lambda x_j) \le (1-x_j)\beta \\ x_j(\bar{a}^T\bar{x} - \mu(1-x_j)) \le x_j\beta \\ x_j(1-x_j) = 0 \end{cases}\right\} \text{ parts of } M_j^{NL}(K).$$

Adding them gives $\bar{a}^T\bar{x} - (\lambda+\mu)\underbrace{(\cancel{x_j}(1-x_j))}_{=0} \le \beta$.

$\Rightarrow$ $\bar{a}^T\bar{x} \le \beta$ is valid for $M_j^{NL}(K)$, and hence for $P_j(K)$.

$\square$

# Disjunctive Programming (DP) and Disjunctive Convexification




Disjunctive Convexification: take intersection of K with a disjunction, and take convex hull. The idea is to obtain a sharper formulation in the process.

A **disjunctive program** (DP) is an optimization problem of the following form:

$$\left\{\begin{array}{ll} \max & \bar{c}^T\bar{x} \\ \text{s.t.} & \bar{x}\in K \\ & \bar{x}\in D_1\cap D_2\cdots\cap D_p \end{array}\right\} \quad (DP)$$

where $K=\{\bar{x}\in\mathbb{R}^n \mid A\bar{x}\le\bar{b}\}$ and

$$D_i = D_{i_1}\cup D_{i_2}\cdots\cup D_{i_{k_i}},\quad i=1,\dots,p,$$ i.e., the

$i^{th}$ disjunction has $k_i$ alternatives.

The set $K$ is the LP-relaxation of (DP).

$D_{i_\ell}$ are polyhedra $(\ell=1,\dots,k_i)$, and are called the terms in the $i^{th}$ disjunction.

## Examples

1. $k_i = 2\ \forall i$, $D_{i_1} = \{\bar{x} \in \mathbb{R}^n \mid x_i = 0\}$ and $D_{i_2} = \{\bar{x} \in \mathbb{R}^n \mid x_i = 1\}$.
   $K = \{\bar{x} \in \mathbb{R}^n \mid A\bar{x} \le \bar{b}\}$ includes the bounds $0 \le x_i \le 1$ for $i = 1, \ldots, p$, $p \le n$.
   Then (DP) is the usual 0-1 (M)IP. if $p < n$, we get MIP.

2. $k_i = 2\ \forall i$, $D_{i_1} = \{\bar{x} \in \mathbb{R}^n \mid x_{\ell_1} = 0\}$ and $D_{i_2} = \{\bar{x} \in \mathbb{R}^n \mid x_{\ell_2} = 0\}$,
   while $K$ contains bounds $x_\ell \geq 0\ \forall \ell$. Here (DP) is a linear program with complementarity constraints of the form $x_{\ell_1} x_{\ell_2} = 0$.

We can solve DP "easily" if we have all inequalities for $\text{conv}\left[K \cap (D_1 \cap \cdots \cap D_p)\right]$. But when do we get efficient representations?

**Notation** $A \cap_c B = \text{conv}(A \cap B)$.

Assuming we know the sharp representation of $K \cap D_i\ \forall i$, we can devise a theoretical cutting plane algorithm for DP.

## Theoretical Cutting Plane Algorithm

**Step 0** $K_0 = K = \{\bar{x} \mid A\bar{x} \le \bar{b}\}$

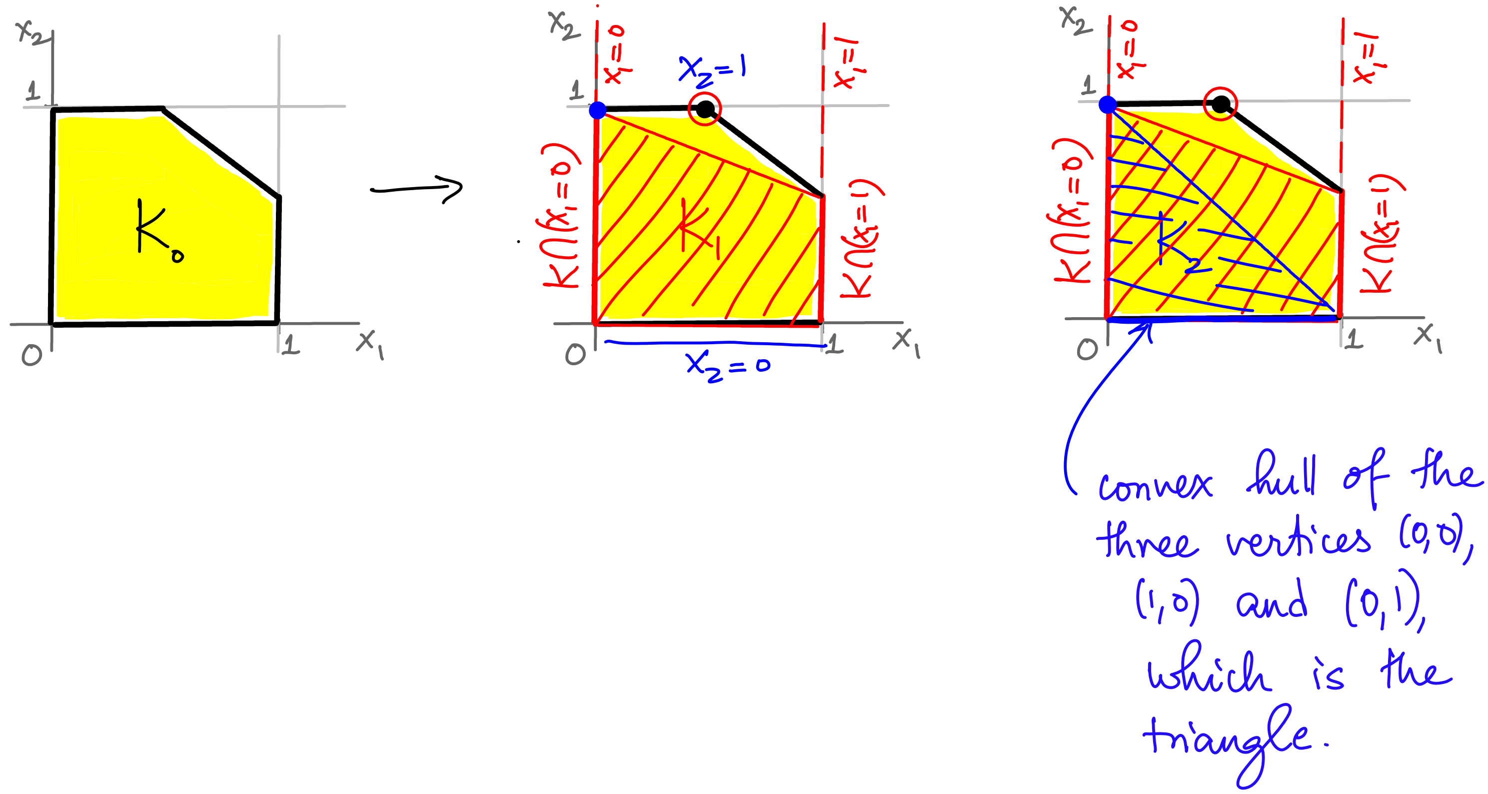
**Step $i$** $(1 \le i \le p)$

if $K_{i-1} = \{\bar{x} \mid A_{i-1}\bar{x} \le \bar{b}^{i-1}\}$

set $K_i = K_{i-1} \cap_c D_i = \text{conv}\left[K_{i-1} \cap D_i\right]$; generate all inequalities for $K_{i-1} \cap_c D_i$

$i \leftarrow i+1$;

**Example** $D_1 = \{x_1 = 0 \vee x_1 = 1\}$, $D_2 = \{x_2 = 0 \vee x_2 = 1\}$.

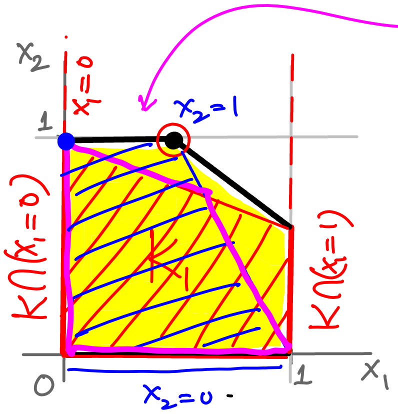



Here, $K_2 = \text{conv}[K \cap D_1 \cap D_2] = K \cap_c (D_1 \cap D_2)$.

Hence, this is a "good instance".

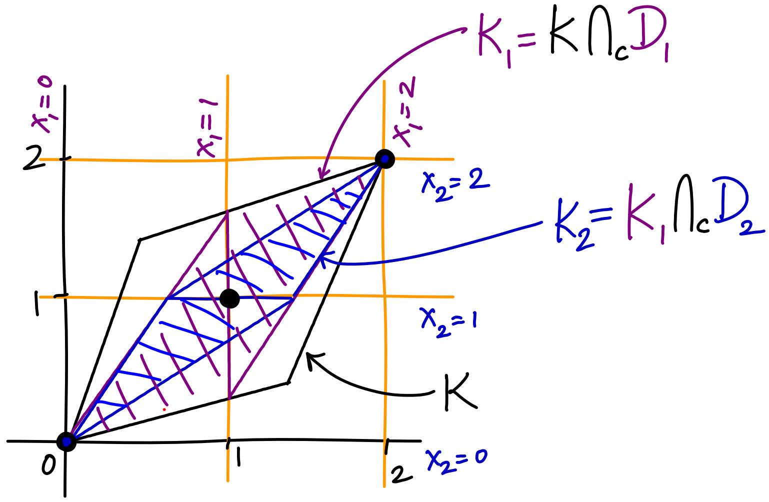
**Remark** $\underbrace{\text{conv}[K \cap D_1] \cap \text{conv}[K \cap D_2]} \neq \text{conv}[K \cap D_1 \cap D_2]$!

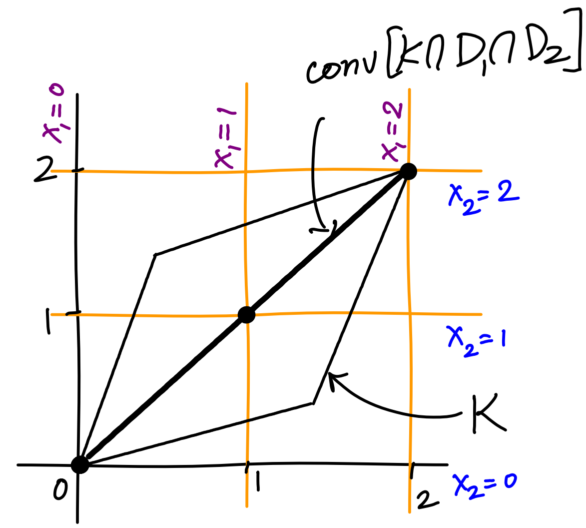
(typically)



## A "bad" instance $p=2$, $k_i=3$, $i=1,2$

$$D_i = (x_i=0)\vee(x_i=1)\vee(x_i=2), \quad i=1,2.$$




Here, $K_2 \neq \text{conv}[K\cap D_1 \cap D_2]$.

Q: When is $K_p = K\cap_c (D_1\cap\cdots\cap D_p)$, where

$$K_p = (\cdots((K\cap_c D_1)\cap_c D_2)\cdots \cap_c D_p)\ ?$$

In general '=' does not hold above.

Notice that in Example 1, the disjunctions $(x_1=0)\vee(x_1=1)$ and $(x_2=0)\vee(x_2=1)$ both defined faces of $K$, while this was not the case in Example 2 ($x_1=1$ and $x_2=1$ both did not define faces of $K$). It turns out that if all terms in each disjunction defines a face of $K$, things are nice!

# MATH 567: Lecture 21 (03/27/2025)

* facial disjunctions
* practical algorithm
* rank of cuts
* semidefinite relaxation

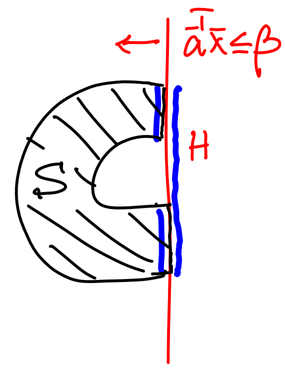
**Def** $D_i$ is a **facial disjunction** w.r.t. $K$ if $D_{ij}$ are faces of $K$ for $j=1,\dots,k_i$, i.e., $D_{ij} = \{\bar{x} \mid (\bar{a}^{ij})^T\bar{x} = \beta_{ij}\}$ where $(\bar{a}^{ij})^T\bar{x} \leq \beta$ is a supporting hyperplane of $K$.

**Theorem 16** If $D_1,\dots,D_p$ are facial disjunctions, then

$$K_p = K \cap_c (D_1 \cap \cdots \cap D_p)$$

in the theoretical algorithm.

**Proposition 17** Let $S$ be any set (possibly nonconvex), and $H = \{\bar{x} \mid \bar{a}^T\bar{x} = \beta\}$ is a hyperplane such that $\bar{a}^T\bar{x} \leq \beta \;\forall \bar{x} \in S$. Then $H \cap \text{conv}(S) = \text{conv}(H \cap S)$.




**Proof (Theorem 16)**

We show the result for $p=2, k_i=2, i=1,2$.

We need to show

$$(K \cap_c D_i) \cap_c D_j = K \cap_c (D_i \cap D_j)$$

$$(K \cap_c D_i) \cap_c D_j = \text{conv}\left[\text{conv}(K \cap D_i) \cap D_j\right] \qquad D_j = D_{j_1} \cup D_{j_2}$$

$$= \text{conv}\left[\left(\text{conv}(K \cap D_i) \cap D_{j_1}\right) \cup \left(\text{conv}(K \cap D_i) \cap D_{j_2}\right)\right]$$

$$= \text{conv}\left[\text{conv}(K \cap D_i \cap D_{j_1}) \cup \text{conv}(K \cap D_i \cap D_{j_2})\right]$$

by Proposition 17.

$$= \operatorname{conv}\left[(K \cap D_i \cap D_{j_1}) \cup (K \cap D_i \cap D_{j_2})\right]$$

$$= \operatorname{conv}\left[K \cap D_i \cap D_j\right] = K \cap_c (D_i \cap D_j). \qquad \square$$

The theoretical algorithm might not work well in practice. Getting efficient descriptions of the convex hulls in each step might be difficult.

## A practical Algorithm

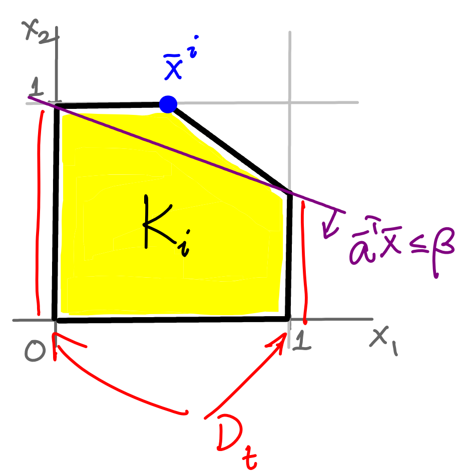
Step 0: $K_0 = K = \{\bar{x} \mid A\bar{x} \leq \bar{b}\}$

Step $i$ $(i \geq 1)$ : $K_i$ : current relaxation, and
$\bar{x}^i$ : optimal solution over $K_i$

Find $\bar{a}^T\bar{x} \leq \beta$ such that

1. $\bar{a}^T\bar{x} \leq \beta$ is valid for $K_i \cap D_t$ for some $t > i$.
2. $\bar{a}^T\bar{x} \leq \beta$ is violated by $\bar{x}^i$ (i.e., $\bar{a}^T\bar{x}^i > \beta$).

Set $K_{i+1} \leftarrow$ LP relaxation of $K_i \cap D_t \cap \{\bar{x} \mid \bar{a}^T\bar{x} \leq \beta\}$.




How do we find $(\bar{a}, \beta)$?

Solve an LP with $\bar{a}, \beta$ as variables.

$$\max \quad \beta - \bar{a}^T \bar{x}^i$$ is given

s.t. ( $\bar{a}^T\bar{x} \leq \beta$ is derivable from $K_i \cap D_t$ by Farkas' Lemma ) ——— (1)

normalization constraint : e.g.,

$$\sum_{i=1}^{n} |a_i| + |\beta| \leq 1.$$ → we can linearize this constraint

Without the normalization, the LP could be unbounded. If $(\beta, \bar{a}^T)$ works, then $100\beta - 100\bar{a}^T\bar{x}^i$ gives a bigger separation. And so does $10000\beta - 10000\bar{a}^T\bar{x}^i$.

For (1), we will use variables representing the multipliers for deriving the constraint $\bar{a}^T\bar{x}^i \leq \beta$ from $K_i \cap D_t$.

Q. How good is any such cutting plane algorithm? First, we define rank of cuts.

## Rank of cuts

$K = \{\bar{x} \mid A\bar{x} \leq \bar{b}\}$ : All inequalities in $A\bar{x} \leq \bar{b}$, or derivable from $A\bar{x} \leq \bar{b}$ are **rank 0 cuts** (or inequalities).

Let $\bar{a}^T\bar{x} \leq \beta$ be valid for $K$. Then the CG cut $\lfloor \bar{a}^T \rfloor \bar{x} \leq \lfloor \beta \rfloor$ is a **rank 1 CG cut**.

Combining some (or all) rank-1 CG cuts and applying the CG procedure again gives me a **rank 2 CG cut**.

Similar notion of rank can be defined for the LS-procedure, MIG cuts, etc.

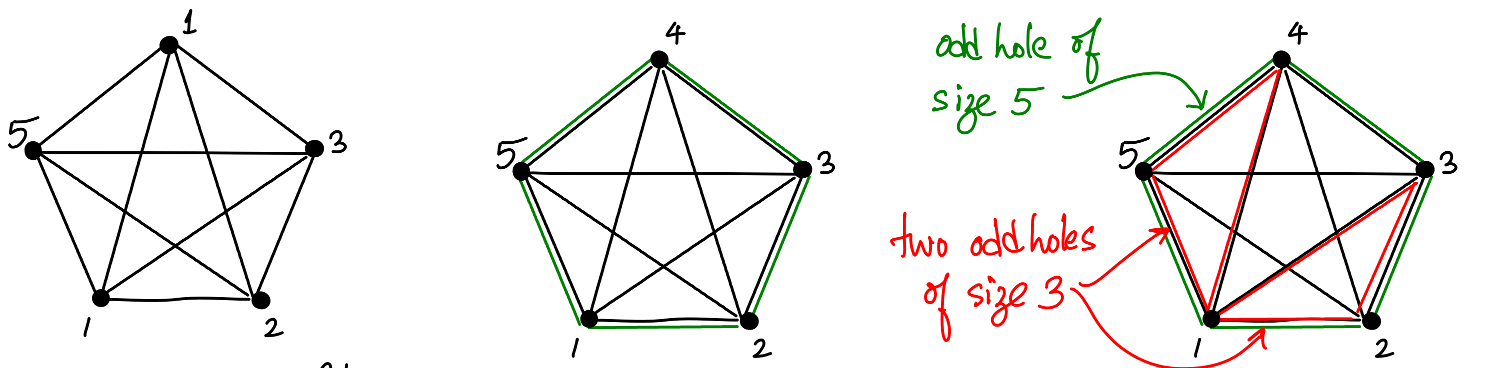
Ideally, we want to derive inequalities with small rank, so that we do not have to apply the cut-procedure too many times.

# Back to LS Procedure

Q: How many steps to generate a good inequality?

Example: node packing on complete graph with 5 nodes.

$$K = \left\{ \begin{array}{l} x_i + x_j \le 1 \ \forall (i,j) \\ 0 \le x_i \le 1 \ \forall i \end{array} \right\}$$




complete graph!

Node packing problem:

A good inequality is $x_1 + x_2 + x_3 + x_4 + x_5 \le 1$ ——— $\circledast$

We can pick at most one node.

Recall the definitions of $M_j^{NL}(K)$, $M_j(K)$, and $P_j(K)$. We have

$M_1^{NL}(K)$ specified as
$$\left\{ \begin{array}{c} (A\bar{x} - \bar{b})x_1 \le 0 \\ (A\bar{x} - \bar{b})(1 - x_1) \le 0 \\ x_1(1 - x_1) = 0 \end{array} \right\}.$$

We get odd hole inequalities of size 3:

$$x_1 + x_2 + x_3 \le 1$$
$$x_1 + x_2 + x_4 \le 1$$
$$\vdots$$

and $x_1 + x_2 + x_3 + x_4 + x_5 \le 2$, odd-hole inequality of size 5.

We do not get ⊛.

In $P_2(P_1(K))$, we get $x_1 + x_2 + x_3 + x_4 \le 1$ —— (2)

To see (2) is valid for $P_2(P_1(K))$, we verify that (2) is valid for $P_1(K) \cap (x_2 = 0 \vee x_2 = 1)$.

$P_1(K) \cap (x_2 = 0)$ gives $x_1 + x_3 + x_4 \le 1$, which is there in $P_1(K)$.

$P_1(K) \cap (x_2 = 1)$ gives $x_1 + x_3 + x_4 \le 0$, which is also valid for $P_1(K)$, since we originally (in $K$ itself) have $x_i + x_j \le 1 \ \forall (i,j)$. Hence $x_2 = 1$ immediately forces $x_j = 0 \ \forall j \ne 2$.

But we still do not get ⊛ in $P_2(P_1(K))$.

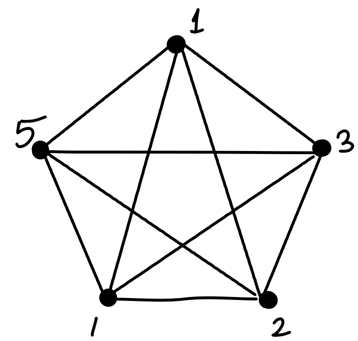
$P_3(P_2(P_1(K)))$ gives $x_1 + \cdots + x_5 \le 1$ —— ⊛ !

$\Rightarrow$ LS-rank of ⊛ is 3.

In general, LS rank of $x_1 + \cdots + x_k \le 1$ is $\le k-2$, and is often $= k-2$.

Q. Could we derive $x_1+x_2+x_3+x_4+x_5 \le 1$ — (*)
in one step? As a rank-1 cut.

YES!




# Semidefinite Relaxation

Recall:

$$M_j^{NL}(K) \equiv \left\{ \begin{array}{c} (A\bar{x}-\bar{b})x_j \le \bar{0} \\ (A\bar{x}-\bar{b})(1-x_j) \le \bar{0} \\ x_j(1-x_j) = 0 \end{array} \right\}, \quad j=1,\dots,n.$$

We linearize $M_j^{NL}(K)$ $(x_j^2 \leftarrow x_j,\ x_ix_j \leftarrow y_i)$ to $M_j(K)$.

Define $M^{NL}(K) = \bigcap_{j=1}^{n} M_j^{NL}(K)$, and $M(K) = \bigcap_{j=1}^{n} M_j(K)$

We could linearize all systems, and then take their intersection. Equivalently, we could take the intersection of the nonlinear systems, and then linearize.

Let $M_+(K) \equiv M(K) \cap \left\{ \bar{x} \;\middle|\; \underline{(a_0+\bar{a}^T\bar{x})^2 \ge 0} \;\; \forall \begin{bmatrix} a_0 \\ \bar{a} \end{bmatrix} \in \mathbb{R}^{n+1} \right\}.$

$\begin{bmatrix} 1 & \bar{x}^T \\ \bar{x} & \bar{x}\bar{x}^T \end{bmatrix}$ is positive semidefinite

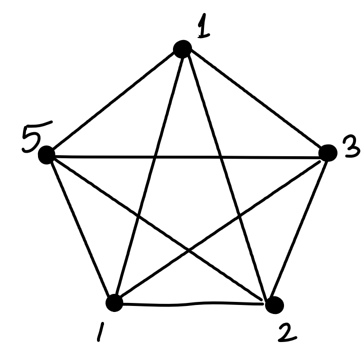
You would think this restriction is always satisfied! But imposing it explicitly makes the difference — see Notes to follow...

$M_+(K)$ is the **semidefinite relaxation** of the problem.

**Def** $A \in \mathbb{R}^{n\times n}$ is **positive semidefinite** (PSD) if $\bar{x}^T A \bar{x} \geq 0 \;\; \forall \bar{x} \in \mathbb{R}^n$. We write $A \succeq 0$

If $A$ is PSD, all its eigenvalues are nonnegative.

Back to example on vertex packing




$M_+(K)$ has $(1-x_1-x_2\cdots x_5)^2 \geq 0$

$$\Rightarrow 1 - 2\sum_{i=1}^{5} x_i + \sum_{i=1}^{5} \underbrace{x_i^2}_{x_i} + 2\sum_{i\neq j} \underbrace{x_i x_j}_{=0} \geq 0$$

$$\Rightarrow 1 - \sum_{i=1}^{5} x_i \geq 0,$$ which is ⊛!

$M(K)$ has $(x_i + x_j \leq 1)x_i \Rightarrow x_i x_j \leq 0$ as $x_i^2 = x_i$

$M(K)$ also has $(-x_j \leq 0)\, x_i \Rightarrow x_i x_j \geq 0$

$\Rightarrow x_i x_j = 0.$

Hence the semidefinite relaxation rank of ⊛ is 1!

Notes

① $M^{NL}(K)$ contains inequalities for

$$X = \begin{pmatrix} 1 & \bar{x}^T \\ \bar{x} & \bar{x}\bar{x}^T \end{pmatrix} = \left[\begin{array}{c|cccc} 1 & x_1 & x_2 & \cdots & x_n \\ \hline x_1 & x_1^2 & & & \\ x_2 & & x_2^2 & & x_i x_j \\ \vdots & & & \ddots & \\ x_n & & x_j x_i & & x_n^2 \end{array}\right].$$

Inequalities for $X$ can be written as

$$B \bullet X \geq 0, \text{ where } A \bullet B = \text{trace}(A^T B)$$

② To get $M(K)$ from $M^{NL}(K)$, we replace $x_i^2$ by $x_i$ and $x_i x_j$ by $y_{ij}$. Hence inequalities of $M(K)$ can be written as

$$C \bullet \begin{pmatrix} 1 & \bar{x}^T \\ \bar{x} & Y \end{pmatrix} \geq 0, \text{ where } \text{diag}(Y) = \bar{x}.$$

③ $(a_0 + \bar{a}^T\bar{x})^2 \geq 0$ can be written equivalently as

$$[a_0 \; \bar{a}^T] \begin{bmatrix} 1 & \bar{x}^T \\ \bar{x} & \bar{x}\bar{x}^T \end{bmatrix} \begin{bmatrix} a_0 \\ \bar{a} \end{bmatrix} \geq 0 \Rightarrow [a_0 \; \bar{a}^T] \begin{bmatrix} a_0 + \bar{a}^T\bar{x} \\ a_0\bar{x} + (\bar{x}\bar{x}^T)\bar{a} \end{bmatrix} = [a_0 \; \bar{a}^T] \begin{bmatrix} (a_0 + \bar{a}^T\bar{x}) \\ (a_0 + \bar{a}^T\bar{x})\bar{x} \end{bmatrix}$$

$$= (a_0 + \bar{a}^T\bar{x})(a_0 + \bar{a}^T\bar{x}) \geq 0.$$

In other words, $M_+(K)$ has $\begin{bmatrix} 1 & \bar{x}^T \\ \bar{x} & Y \end{bmatrix} \succeq 0$ as added constraints.

# MATH 567: Lecture 22 (04/01/2025)

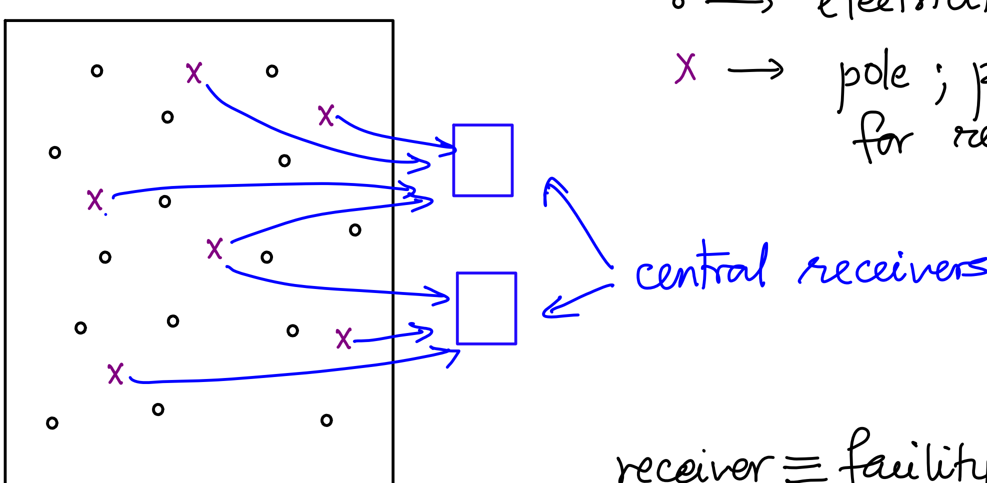
Today: * set covering problem (heuristics)

## Solving Large sized Integer Programs: Set Covering Problem

Given:
1) $n$ customer locations
2) $m$ candidate facility locations
3) for each candidate facility location, the subset of customers that can be covered.

Assume no limits on # customers a facility can serve — hence we could look at it as an instance of uncapacitated facility location (UFL) problem.

<u>Application</u> Receiver location problem for reading electricity/gas meters.




o → electricity meter

x → pole; potential locations for receivers/amplifier

receiver ≡ facility    meter ≡ customer

The meters transmit readings to (at least one) receiver, which amplifies it before transmitting to a central receiver.

Goal: Identify which poles to locate receivers on, so that we minimize the total # recievers (i.e., facilities) used such that every meter (i.e., customer) can transmit to at least one receiver.

Such problems are often quite big, and hence cannot be handled easily as (M)IPs. We consider heuristics.

→ algorithms that are not guaranteed to find the optimal solution.

Also, we do not get any measures of the quality of the solutions found.

But, they often work well in practice!

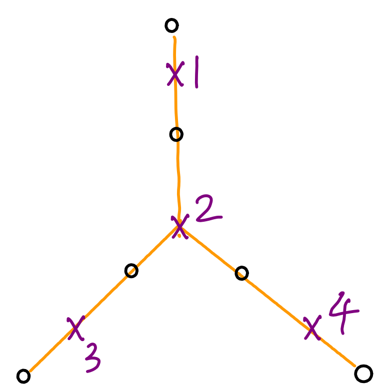
Heuristics

1. **Greedy algorithm**: In each step, pick the pole that covers the **largest number of uncovered meters**.

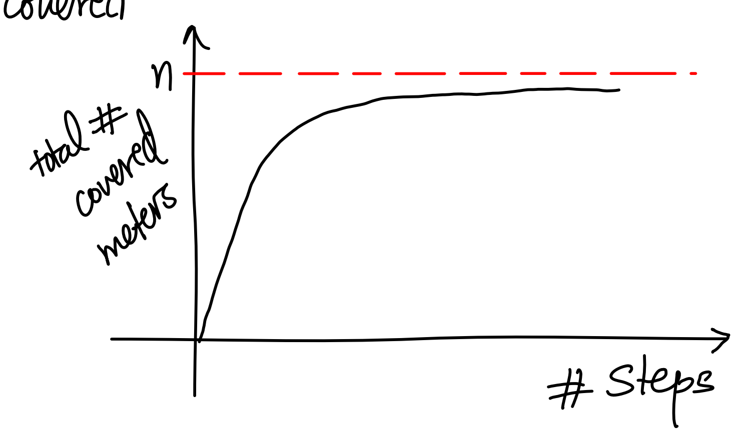
Break ties arbitrarily.

In general, will not give optimal solution.

Here, greedy gives $\{2,1,3,4\}$, while optimal solution is $\{1,3,4\}$.




As the heuristic runs, the #-covered meters "plateaus" out.

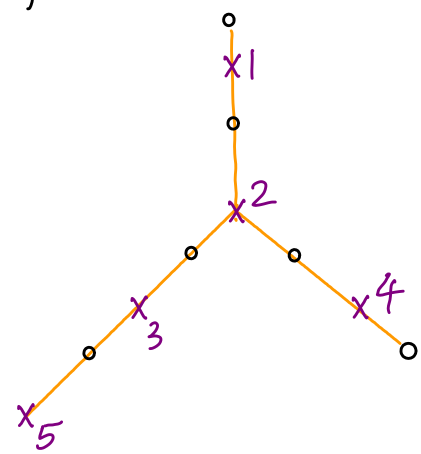


# Cleaning up the solution

If removing the $i^{th}$ pole from the set of selected poles leaves all meters covered (as covered up to now), then remove that pole. Repeat for $i = 1, \dots, p$ after $p$-steps, for all $p$ (or, say, repeat after every $10^{th}$ pole).

Clean up gives optimal solution in the previous example.

But in this example, if greedy gives $\{2, 1, 4, 5\}$, clean up will do nothing.

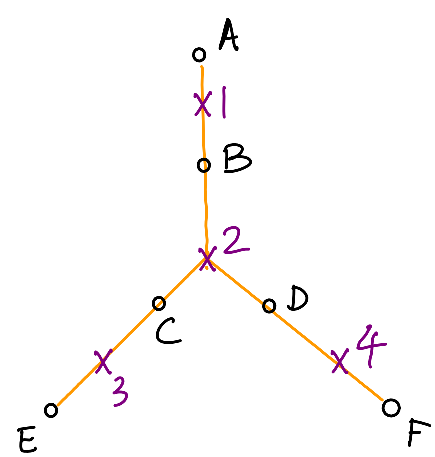



(2) **Modified Greedy algorithm** (Balas and Ho)

- Uses more foresight than greedy
- IDEA: Define a scoring function, and in each step, pick the pole with the largest value.

**Def** A meter is called **hard to cover** if the number of poles that cover it is minimal.

Here, A, E, F are hard-to-cover.



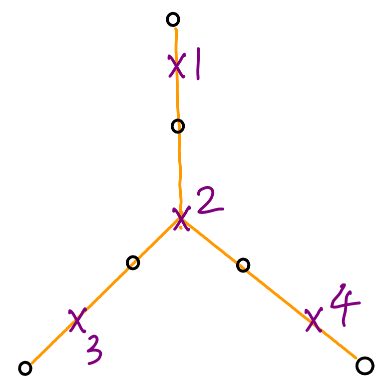
For pole $j$, define

$$\text{Score}_1(j) = \begin{cases} \text{total \# meters covered by pole } j \text{ if it covers at least one hard-to-cover meter} \\ 0, \quad \text{otherwise.} \end{cases}$$

and

$$\text{Score}_2(j) = (\#\text{ meters covered by pole } j) \times (\#\text{ hard-to-cover meters covered by pole } j).$$

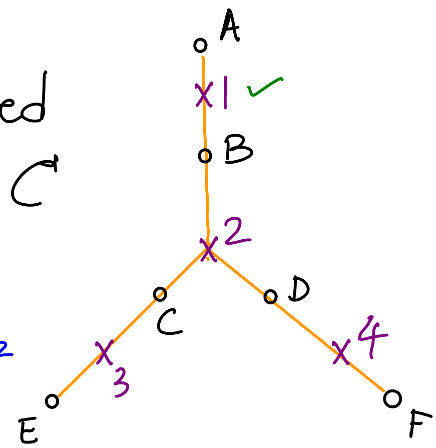
Modified greedy works in this example. $\text{Score}_1(2) = 0$ and $\text{Score}_1(j) = 2$ for $j = 1, 3, 4$, at start, and do not change as the algorithm proceeds. Hence, we select $\{1, 3, 4\}$.




## A Generalized Scoring Function

**Def** Suppose modified greedy at some intermediate step has selected a subset $P' \subseteq P$ of the poles. Let $m_{P'}$ be the minimum # poles that can cover a meter (not yet covered). For $t \in \mathbb{Z}_{>0}$, an uncovered meter is **t-hard to cover** if the # poles that can cover it is $< m_{P'} + t$.

So, 1-hard to cover $(t=1) \equiv$ hard-to-cover as previously defined.




**Example** Let $P' = \{1\}$. Then $m_{P'} = 1$ (for E and F). Meter C is covered by both poles 2 and 3. Hence, C is 2-hard to cover, as $2 < 1 + 2$.

$t=2$ poles 2,3 $m_{P'}$ $t=2$

E, F are 1-hard to cover ($t=1$).

C, D are <u>also</u> $t$-hard to cover for all $t \geq 3$.

Let $s(j,t)$ = # $t$-hard to cover meters covered by pole $j$.

We define

$$Score_g(j) = s(j,\infty) \prod_{t=1}^{k} [s(j,t)]^{1/t}$$

weighs 1-hard-to-cover meters
> 2-hard-to-cover meters
> 3-hard-to-cover
⋮

general

total # uncovered meters covered by $j$

where $k \geq 1$ is a fixed positive integer.

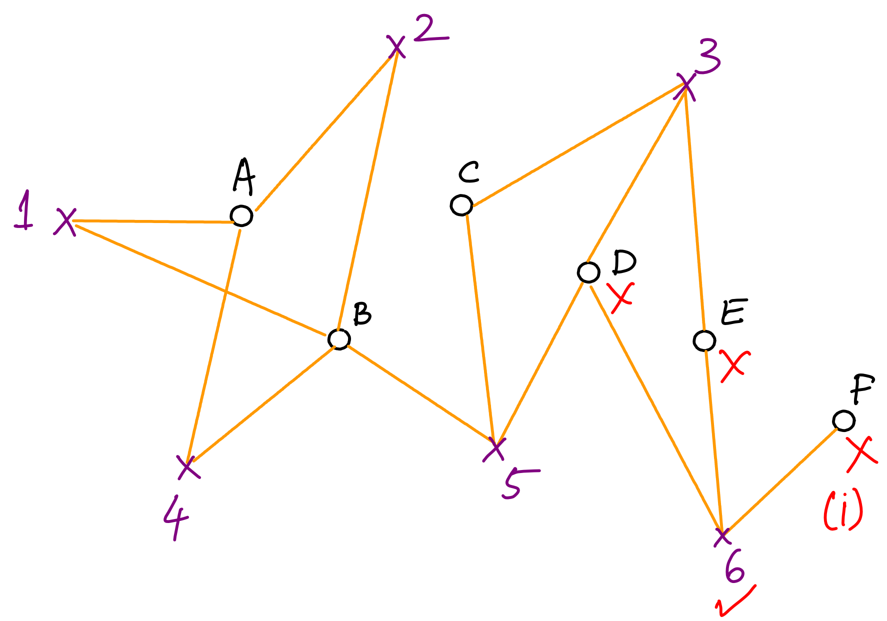
<u>Note</u>

1. $Score_g(j) = Score_2(j)$ when $k=1$.
2. $Score_g(j) = Score_1(j)$ if there is a unique meter covered by pole $j$ that is 1-hard to cover
3. higher $k \Rightarrow$ more foresight

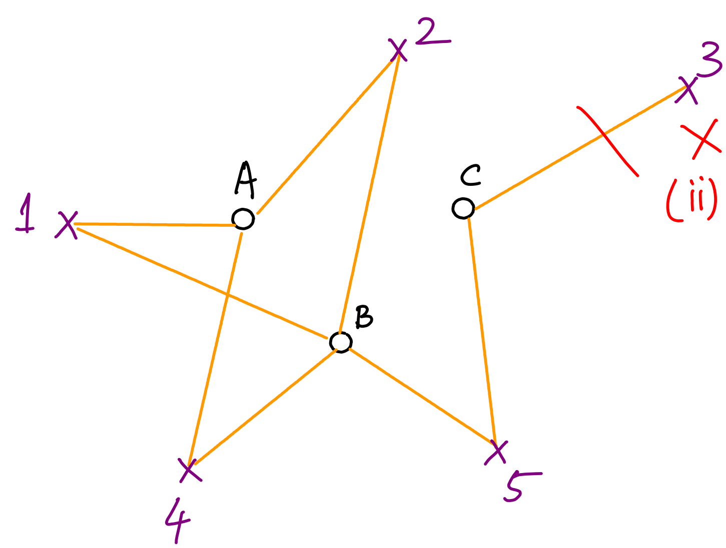
## (3) Preprocessing

Reduces the size of the problem, but does not typically give an optimal solution (except in trivial cases).

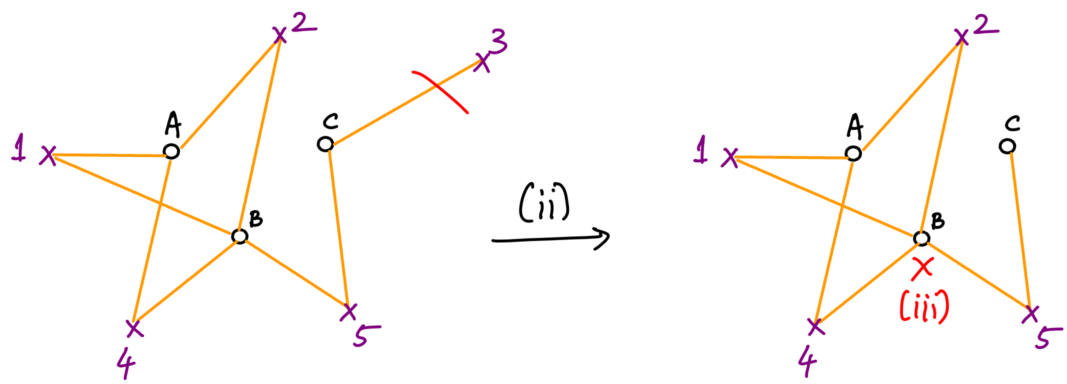
We illustrate the main steps on an example.



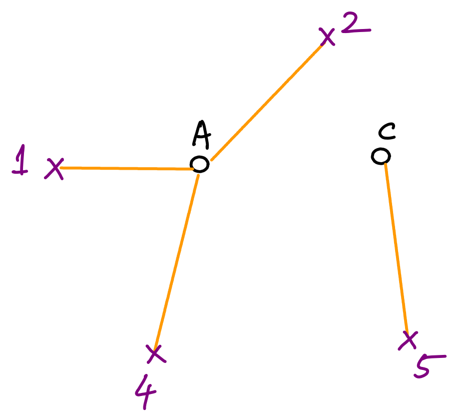

(i) Only 6 covers F $\Longrightarrow$ choose 6, delete D, E, F (as pole 6 covers D, E, F).



(ii) Now pole 5 covers all meters that pole 3 covers $\Rightarrow$ delete pole 3.




(iii) $\{1,2,4,5\}$ and $\{A,B,C\}$ are left. Now, poles covering A also cover B $\Longrightarrow$ delete B.




another application of step(i) here!

(iv) Left with $\{1,2,4,5\}$ and $\{A,C\}$.
Optimal solution: pick 5 (as only 5 covers C), and pick one out of 1,2,4, say, 1 $\Longrightarrow$ $\{1,5\}$.
(On larger instances, we run greedy/modified greedy on this smaller instance).

(v) Extend (optimal) solution in Step (iv) to an (optimal) solution to the whole problem by adding pole 6 chosen in Step (i) $\Longrightarrow$ Solution is $\{1,5,6\}$.

# MATH 567: Lecture 23 (04/03/2025)

Today: * Heuristic algo for set cover
* variants of set cover problem

## Receiver location problem: Heuristic algorithm

Let $C_i = \{ j \in P \mid j \text{ covers } i \}$, and $C = \{C_i\}_i$.

coverage info for meter $i$ — all coverage info

$$\text{HEURISTIC\_SETCOVER}(P, M, C, k, \varepsilon)$$

$P$: set of poles; $M$: set of meters; $k \in \mathbb{Z}_{>0}$; $\varepsilon$: tolerance for comparing score

INPUT: set of poles $P$, set of meters $M$, coverage $C$, parameter $k$, tolerance $\varepsilon$;

OUTPUT: $P' \subseteq P$ that covers all $i \in M$.

Initialization: $P' = \emptyset$, $M' = \emptyset$ (where $M' \subseteq M$ is the set of meters covered by $P'$).

while $M' \neq M$ do

  PREPROCESS$(P \setminus P', M \setminus M')$; (set difference)
  ↳ steps (i), (ii), (iii)

  Compute Score$(\cdot)$ for $(P \setminus P', M \setminus M')$; (after preprocessing)

  Select $j \in P \setminus P'$ with $\text{Score}(j) = \max\limits_{\ell \in P \setminus P'} (\text{Score}(\ell)) \pm \varepsilon$;

  $P' = P' \cup \{j\}$;

  Update $M'$;

  CLEANUP $P'$; ← could do CLEANUP only after every $10^{th}$ pole is selected, say.

end-while

## 4. IP formulation

We first look at the IP formulation to describe how to implement the steps in the heuristic algorithm (preprocessing, in particular)

We let $M := \{1,2,\dots,m\}$ denote the set of meters, and $P := \{1,2,\dots,p\}$ the set of poles. The coverage data is specified as $C_i = \{j \in P \mid j \text{ covers } i\}$ $\forall i \in M$.

Let $x_j = \begin{cases} 1 & \text{if pole } j \text{ is selected (to locate a receiver)} \\ 0 & \text{otherwise} \end{cases}$

$\uparrow$ facility

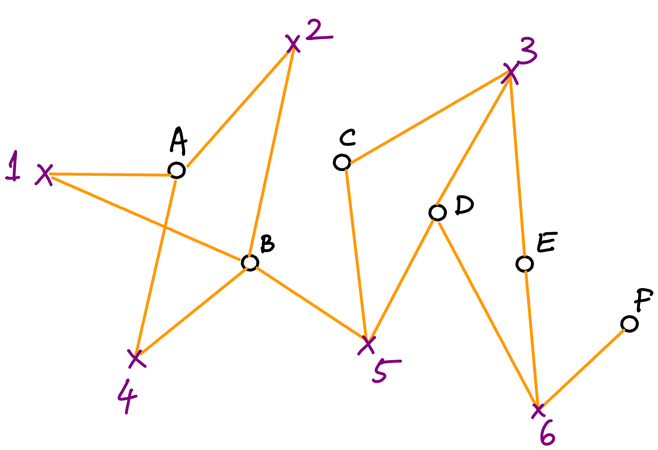
Here is the IP:

$$\min \sum_{j=1}^{p} x_j$$

$$\text{s.t.} \quad \sum_{j \in C_i} x_j \geq 1 \quad \forall i \in M \qquad (\Delta)$$

$$x_j \in \{0,1\} \quad \forall j \in P$$

Illustration on the example:




$$\min \quad x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \qquad \mathbb{1}$$

$$\text{s.t.} \quad \boxed{\begin{array}{l} x_1 + x_2 + x_4 \\ x_1 + x_2 + x_4 + x_5 \\ x_3 + x_5 \\ x_3 + x_5 + x_6 \\ x_3 + x_6 \\ x_6 \end{array}} \begin{array}{ll} \geq 1 & (A) \\ \geq 1 & (B) \\ \geq 1 & (C) \\ \geq 1 & (D) \\ \geq 1 & (E) \\ \geq 1 & (F) \end{array}$$

$A\bar{x} \rightarrow$

$$x_j \in \{0,1\}, \; j = 1, \dots, 6.$$

$A\bar{x} \geq \mathbb{1}$ $\searrow$ vector of ones

# Steps in Preprocessing

$$
\begin{array}{lllr}
 & & (ii) & \\
\min & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 & & \\
\text{s.t.} & x_1 + x_2 \quad + x_4 & \geq 1 & (A) \\
 & x_1 + x_2 \quad + x_4 + x_5 & \geq 1 & (B) \ (iii) \\
 & x_3 \quad + x_5 & \geq 1 & (C) \\
A\bar{x} & x_3 \quad + x_5 + x_6 & \geq 1 & (D) \\
 & x_3 \quad + x_6 & \geq 1 & (E) \ (i) \\
 & 1 = x_6 & \geq 1 & (F)
\end{array}
$$

$$x_j \in \{0,1\}, \ j \in \{1,\ldots,6\}$$

(i) (singleton rows): Set $x_6 = 1$, delete all rows containing $x_6$, i.e., (D) & (E)

(ii) After Step (i), $x_5$-column dominates $x_3$-column $\Rightarrow$ set $x_3 = 0$.

**Def** For $\bar{u}, \bar{v} \in \{0,1\}^n$, $\bar{u}$ (strictly) dominates $\bar{v}$ if $u_i \geq v_i \ \forall i$, and $u_j > v_j$ for at least one $j$.

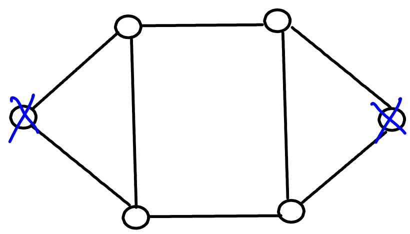
(iii) Now, row (B) dominates row (A) $\Rightarrow$ eliminate row (B).

Thus, all operations are reduced to matrix operations.

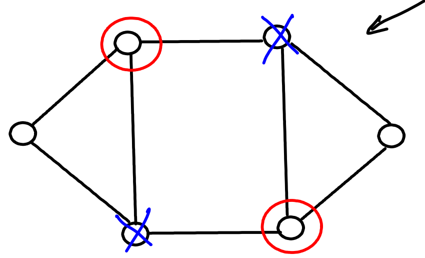
# Variants of Set Covering

**Variant 1**: 'o' = 'x' (facility and customer locations are same)

Consider the following instance:



Here is an optimal solution with size 2.



But this solution is better, as two customers are covered twice (indicated by ⊙).

This example motivates the following variant of the default set covering problem:

**Objectives**: (1) minimize total # facilities;
(2) among all optimal solutions to (1), pick one that maximizes # customers covered two or more times.

Define $y_i = \begin{cases} 1 & \text{if customer } i \text{ covered twice or more} \\ 0 & \text{o.w.} \end{cases}$

IP is NOT

$$\min \sum_{j=1}^{n} x_j - \sum_{i=1}^{m} \cancel{y_i}$$

$$\text{s.t.} \quad \sum_{j \in C_i} x_j - y_i \geq 1$$

$$x_j, y_i \in \{0,1\}$$

applies to the general set cover problem where 'x' and 'o' are distinct.

The objective function does not work quite correctly!

If $\bar{x}, \bar{y}$ is an optimal solution to this IP, we can find $\bar{x}', \bar{y}'$ with $\sum_{j \in C_i} x_j' = \sum_{j \in C_i} x_j + 1$ and

$$\sum_{i=1}^{m} y_i' \geq \sum_{i=1}^{m} y_i + 2.$$

Hence, we use the following objective function instead:

$$\min \sum_{j=1}^{p} x_j - \underbrace{\varepsilon \sum_{i=1}^{m} y_i}_{} \quad \text{where } \varepsilon = \frac{1}{m+1}.$$

The contribution of the second sum term is hence $< 1$.

<u>Variant 2</u> Let $w_i$ = importance of covering customer $i$ ($w_i \geq 0$) and let $p_0$ be the maximum number of facilities (poles picked)

($p_0 \in \mathbb{Z}_{>0}$ represents a budget constraint).

<u>Objective</u>: Maximize sum of importance of covered customers.

Let $s_i = \begin{cases} 1 & \text{if customer } i \text{ is covered,} \\ 0 & \text{otherwise.} \end{cases}$ $i \in \{1, \dots, m\} = M$.

it is not required to cover all customers

IP:

$$\max \sum_{i=1}^{m} w_i s_i$$

$$\text{s.t.} \quad \sum_{j \in C_i} x_j \geq s_i \quad \forall i \in M$$

$$\sum_{j=1}^{p} x_j \leq p_0$$

$$x_j, s_i \in \{0,1\}$$

Variant 3 (Existing and new facilities)

$P = \{1, 2, \dots, p_{ex}, p_{ex}+1, \dots, p_{ex}+p_{new}\}$ $\quad p_{new} = \#$ new facilities
$p_{ex} = \#$ existing facilities

Objectives: (1) minimize total # facilities;
(2) among all optimal solutions to (1), pick one that minimizes # new facilities.

We want to use as many of the existing facilities first, before using new ones. We could also incorporate the costs for establishing the new facilities. It is not necessary to use all existing facilities — or, there could be a cost for using one.

$$\min \quad \sum_{j=1}^{p_{ex}} x_j + (1+\varepsilon) \sum_{j=p_{ex}+1}^{p_{ex}+p_{new}} x_j$$

$$\text{s.t.} \quad \sum_{j \in C_i} x_j \geq 1 \quad \forall i \in M$$

$$x_j \in \{0,1\},$$

$$\text{where } 0 < \varepsilon \leq \frac{1}{p_{new}+1}$$

Need $0 < \varepsilon \leq \frac{1}{p_{new}+1}$, in the same way as in Variant 1, so that the contribution of the extra sum term is $< 1$.

# MATH 567: Lecture 24 (04/08/2025)

Today: * p-center and p-median problems

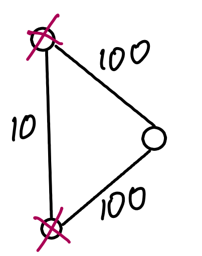
## p-center and p-median problems

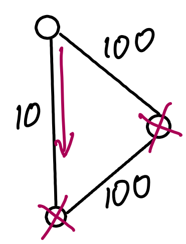
IDEA: Locate at most $p$ facilities, assign every customer to one facility

minimize
- → total distance of all customers to their facility } p-median
  warehouses
- → maximum distance of any one customer to their facility } p-center
  fire stations

Consider this instance:
(numbers on edges are $d_{ij}$)

distance between nodes $i$ & $j$




objective function values

| 2-center | 2-median |
|---|---|
| 100 | 100 |
| 10 | 10 |

## IP formulation

$$x_j = \begin{cases} 1 & \text{if facility located at } j, \; j \in N = \{1, 2, \ldots, n\} \\ 0, & \text{otherwise, and} \end{cases}$$

$$y_{ij} = \begin{cases} 1 & \text{if customer } i \text{ is assigned to facility } j, \; i, j \in N \\ 0, & \text{otherwise.} \end{cases}$$

## Constraints

$$\sum_j y_{ij} = 1 \quad \forall i \qquad \text{(customer } i \text{ assigned to one facility)}$$

$$\sum_j x_j \leq p \qquad \text{(at most } p \text{ facilities installed)}$$

$$\left( y_{ij} \leq x_j \quad \forall i, j \right) \qquad \text{(customer } i \text{ can be assigned to } j \text{ only if facility located there)}$$

→ disaggregated constraints

or

$$\left( \sum_{i=1}^{n} y_{ij} \leq n x_j \quad \forall j \right)$$

→ aggregated constraints

$$x_j, y_{ij} \in \{0,1\}$$

## Objective functions

$$\min \quad z = \sum_j \sum_i d_{ij} y_{ij} \qquad \text{(}p\text{-median)}$$

$$\left. \begin{array}{l} \min \quad z \\ z \geq d_{ij} y_{ij} \quad \forall i, j \end{array} \right\} \qquad \text{(}p\text{-center)}$$

$d_{ij}$ = distance between nodes $i$ and $j$

To also minimize # poles ($\leq p$) among optimal solutions to $p$-center/$p$-median, we can set

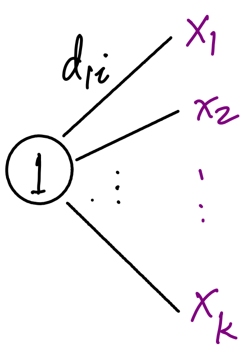
$$\min \quad z + \varepsilon \sum_{j=1}^{n} x_j \qquad \text{where} \quad 0 < \varepsilon < \min_{i,j} \{d_{ij}\} / n$$

**Applications** $p$-median : warehousing → minimize the total distance from warehouses to all retailers.

$p$-center : firestations → minimize the largest distance of any customers from their assigned fire station.

**Claim** It is enough to put $0 \le y_{ij} \le 1$, i.e., $y_{ij} \in \{0,1\}$ is not needed.

**Proof** Suppose $y_{11} + y_{12} + \cdots + y_{1k} = 1$ and $y_{1j} = 0$ for $j > k$, where $0 < y_{1i} < 1$ for $i = 1, \ldots, k$. So, customer 1 is partially assigned to facilities $1, \ldots, k$.




Then we can make $y_{1,i^*} = 1$ where $d_{1,i^*} = \min\limits_{1 \le i \le k} \{d_{1i}\}$, and set $y_{1i} = 0$ for $i \ne i^*$.

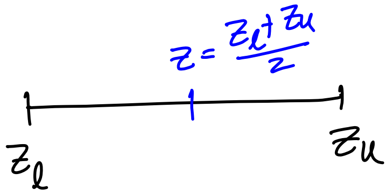
Hence, there always exists an optimal solution with $y_{ij} \in \{0,1\}$. $\square$

$x_j$'s still need to be set as binary variables.

We will look at heuristics for the p-center and p-median problems. The MIP formulations are often too big to handle, even for moderately sized problems.

# Binary Search Heuristic for p-center

Let $z_\ell = 0$, $z_u = M$ (lower, upper bounds on $z$, the objective function)




while ($|z_u - z_\ell| > \varepsilon$) → this $\varepsilon$ is not related to the $\varepsilon$ we used to scale secondary objective function (two pages ago)!

{

$$z = \frac{z_\ell + z_u}{2};$$

$$C_i = \{ j \mid d_{ij} \le z \}$$

Solve a set covering problem (SCP) with $C_i$'s;
↳ use greedy/modified greedy

if optimal value (SCP) $> p$

set $z_\ell = z$; → ignores lower half of $[z_\ell, z_u]$; as we need to increase the allowed distances

else

set $z_u = z$; → ignore upper half of $[z_\ell, z_u]$; We are able to satisfy requirements with $\le p$ facilities, so we could try to tighten the distances now.

end

} (end_while)

# Greedy heuristic for the p-median problem

If $\boxed{p=1,}$ we can locate the one facility optimally.

Let $z_j = \sum_{i=1}^{n} d_{ij}$, pick $j$ with minimal $z_j$.

$\boxed{(p-1) \longrightarrow p:}$ Assuming the first $(p-1)$ facilities stay, we locate the $p^{th}$ facility optimally.

Let $X_{p-1}$ be the set of $(p-1)$ facilities already located.

We compute

$$z_j = \sum_{i} \text{dist}(i, X_{p-1} \cup \{j\}) \quad \text{where}$$

$$\text{dist}(i, X_{p-1} \cup \{j\}) = \min_{k \in X_{p-1} \cup \{j\}} \{d_{ik}\}, \quad \text{and}$$

Select $j$ with the smallest $z_j$, set $X_p = X_{p-1} \cup \{j\}$.

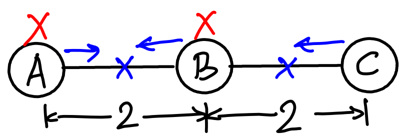
In other words, we are choosing the $p^{th}$ facility greedily.

We could do a clean-up type run through the selected facilities after each facility is picked, or after, say, every $10^{th}$ facility is picked.

# Absolute $p$-center problem

Here, we can locate facilities on the edges of the graph, in addition to locating on its nodes.

Note: The original version is called the vertex $p$-center problem.




Objective function values ($z$):

| $p$ | vertex | absolute |
|---|---|---|
| 1 | 2 | 2 |
| 2 | 2 | 1 |

How do we solve the absolute $p$-center problem? It appears that there could be infinitely many new candidate facility locations!

Even though there might be infinitely many possibilities for the absolute $p$-center problem, we can show the following result.

We can select finitely many points, $\mathcal{N}$, on the edges and then

Absolute $p$-center problem on $G = (V, E)$ $\equiv$ vertex $p$-center on $G' = (V \cup \mathcal{N}, E')$, where

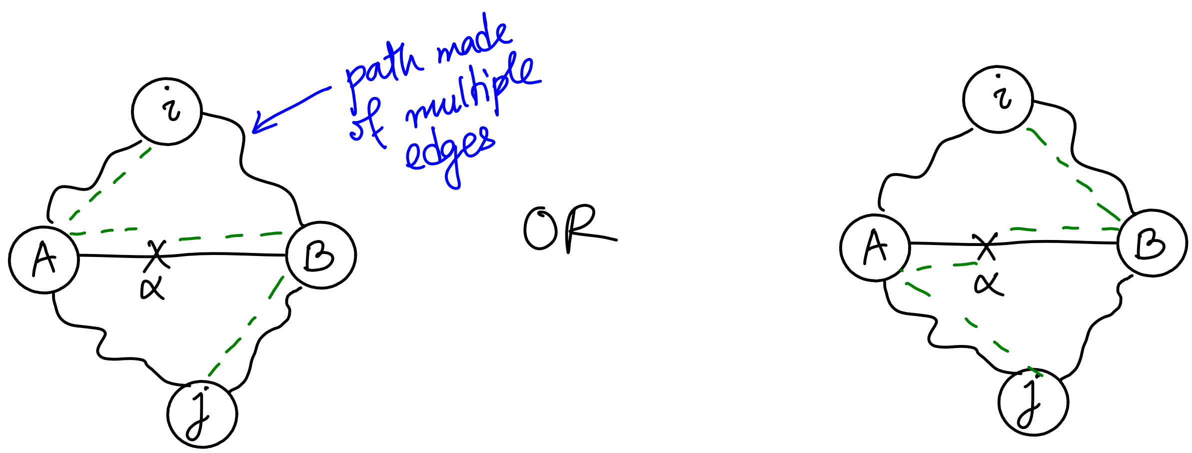
$E'$ consists of original edges in $E$ split into two or more edges when vertices from $\mathcal{N}$ are added.

**Def** A facility $\alpha$ is a **local center** for nodes $i$ and $j$ if

$$d(i,\alpha) = d(j,\alpha) \leq d(k,\alpha) \ \forall\, k \neq i, j, A, B, \alpha.$$

where $\alpha$ is located on $\overline{AB}$, and $i, j, k$ are assigned to $\alpha$.

The shortest paths from $\alpha$ to $i$ and $j$ must go alternatively through $A$ and $B$, as shown below.




**Claim** There exists an optimal solution to the absolute $p$-center problem where every facility is a local center for some $i$ and $j$.

# MATH 567: Lecture 25 (04/10/2025)

Today: * Absolute $p$-center problem
* DKP, lattices, and basis reduction

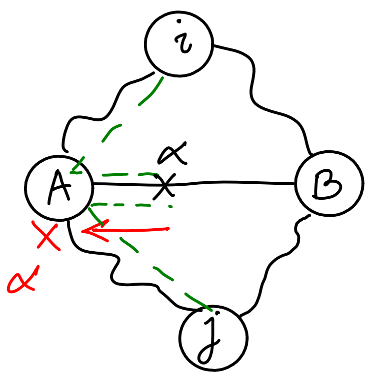
Recall..

**Claim** There exists an optimal solution to the absolute $p$-center problem where every facility is a local center for some $i$ and $j$.

**Proof** Assume a facility $\alpha$ is not a local center.

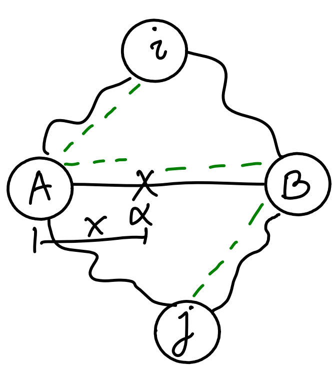
$\Rightarrow d(i,\alpha) > d(k,\alpha)\ \forall k \neq i$, and $k, i$ assigned to $\alpha$.

We can move $\alpha$ closer to $i$ until $d(i,\alpha) = d(k,\alpha)$ for some $k$. We still keep $i$ the farthest node from $\alpha$. This sliding will not worsen the objective function. $\square$




Suppose $\alpha$ is a local center for $i, j$, but the shortest paths from $i$ & $j$ to $\alpha$ both go through $A$, say. Here, we could slide $\alpha$ along $AB$ to $A$, and still maintain $\alpha$ being a local center for $i, j$, while maintaining $d(i,\alpha) = d(j,\alpha)$.

Hence, we should have the following set-up (or its complementary one):




With $d(A,\alpha) = x$, we must have

$$d(i,\alpha) = d(i,A) + x = d(j,\alpha) \leq d(j,A) + x$$

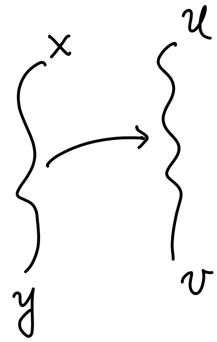
$$\Rightarrow d(i,A) \leq d(j,A).$$

Similarly, we need

$d(i,B) \geq d(j,B)$ here.

**Notation**

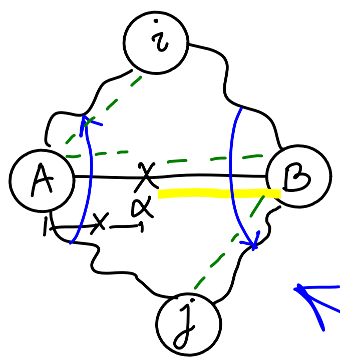
$d(x,y) \geq d(u,v)$



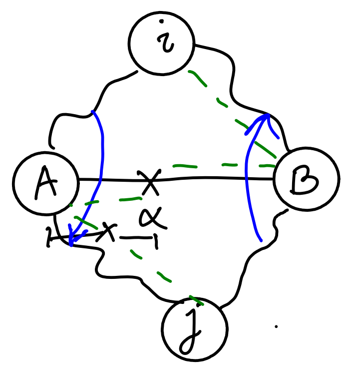

$\equiv$ path $x$-$y$ is greater or equal in length than path $u$-$v$.

## Conditions for existence of a local center for $i, j$ on $\overline{AB}$

**Condition 1:** A local center $\alpha$ can exist on $\overline{AB}$ for $i, j$ if




OR




holds.

with the first option

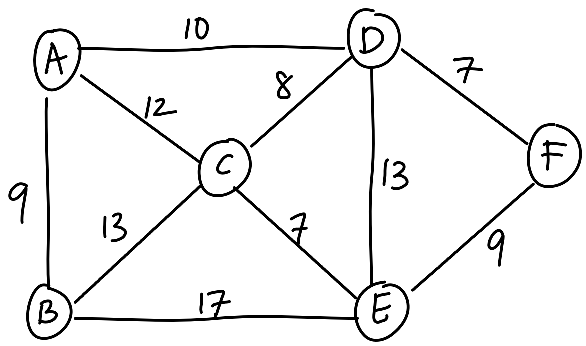
**Condition 2** If condition 1 holds, the location of the local center is given by

$$d(i,A) + x = \big(d(A,B) - x\big) + d(j,B)$$

$$\Rightarrow \quad x = \frac{d(A,B) + d(j,B) - d(i,A)}{2}.$$

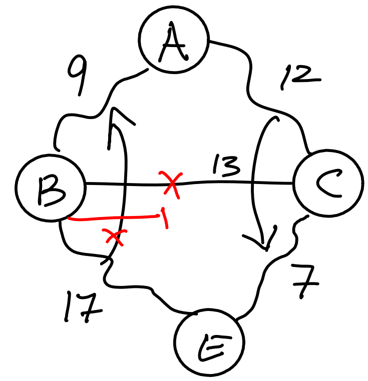
We can derive a similar expression for the second option.

# Example




Identify (possible) local centers:

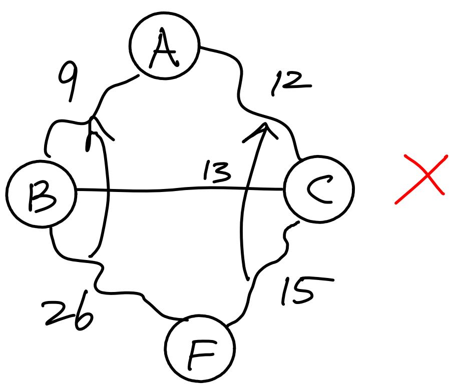
1. On BC




$$9 + x = 7 + (13 - x)$$

$$\Rightarrow x = \frac{11}{2} = 5.5.$$

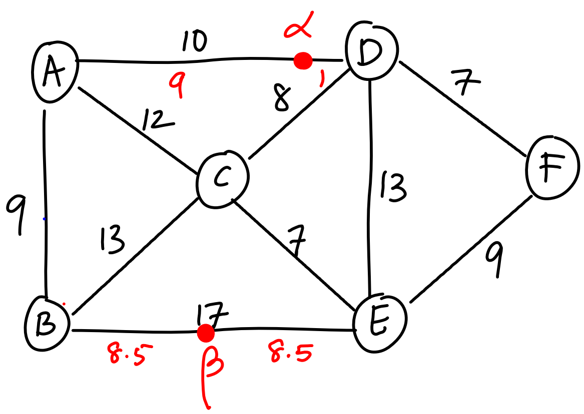
2. Center on BC for $\{i,j\} = \{A,F\}$?



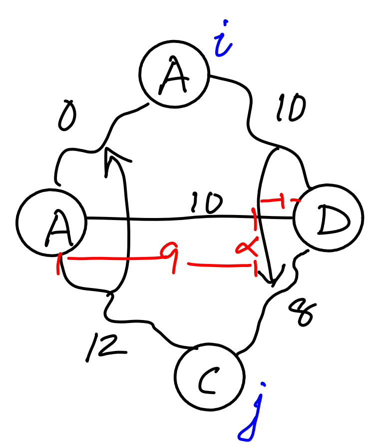

Condition 1 does not hold!

We can check and identify all possible local centers in this manner.

An Optimal solution:




Assignments: $\begin{cases} \alpha : A, C, D, F \\ \beta : B, E \end{cases}$




Optimal $z$ value $= 9$ here.

(for $(A, \alpha)$ and $(C, \alpha)$).

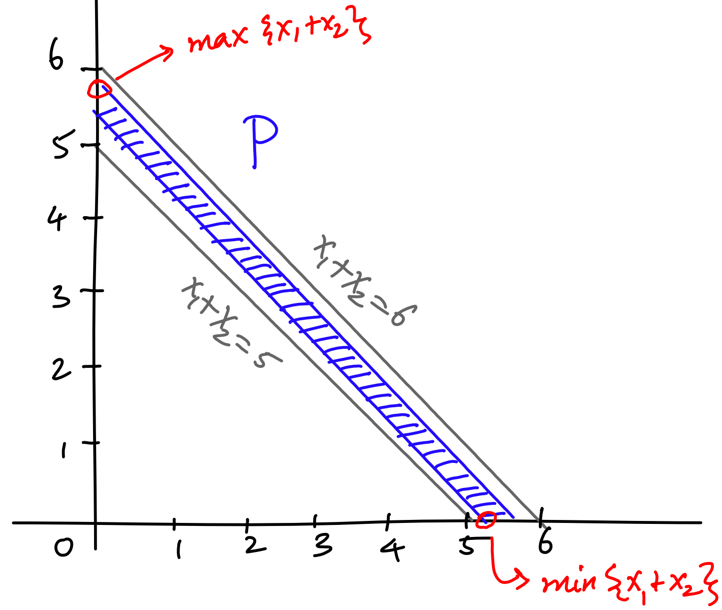
# Lattice basis reduction and IP

Basis reduction (BR) is an important method used in proving several theoretical results in IP, e.g., devising polynomial time algorithms to solve IP when the dimension is considered fixed.

We will introduce the basic concepts related to lattices and BR, and discuss how these concepts help in solving certain classes of IPs that are otherwise hard for the usual methods involving branch-and-cut.

We start by considering a 2D knapsack feasibility problem (KP).

$$(KP)\left\{\begin{array}{l} P\left\{\begin{array}{l} 106 \leq 21x_1 + 19x_2 \leq 113 \\ 0 \leq x_1, x_2 \leq 6 \end{array}\right. \\ x_1, x_2 \in \mathbb{Z} \end{array}\right.$$




Qn: Is $P \cap \mathbb{Z}^2 = \phi$?

Yes, but B&B will take $\approx (6)^2$ nodes to prove this fact when binary branching on $x_1, x_2$ is used.

But consider the direction $x_1 + x_2$.

$$\max\{x_1+x_2 \mid \bar{x} \in P\} = 5.94 \quad \left(\tfrac{113}{19}\right), \text{ and}$$

$$\min\{x_1+x_2 \mid \bar{x} \in P\} = 5.04 \quad \left(\tfrac{106}{21}\right).$$

Hence, branching on $x_1 + x_2$ proves infeasibility at the root node.

This 2D example is an instance of a more general knapsack problem of the form

$$(P)\left\{\begin{array}{c}\beta' \le \bar{a}^T\bar{x} \le \beta \\ \bar{0} \le \bar{x} \le \bar{u}\end{array}\right\} \quad \left.\begin{array}{c} \\ \\ \bar{x} \in \mathbb{Z}^n\end{array}\right\}(KP)$$

where $P \cap \mathbb{Z}^n = \phi$ by design, i.e., (KP) is integer-infeasible, but B&B branching on the individual variables takes an exponential (in $n$) number of BB nodes to prove it.

In particular, we study a class of (KP) problems where $\bar{a} = \bar{p}M + \bar{r}$, with $\bar{p} \ge \bar{0}$, $M >> 1$, and $\bar{p}, \bar{r} \in \mathbb{Z}^n$, $M \in \mathbb{Z}$. Since $\bar{a}$ decomposes in this fashion, we call these problems decomposable knapsack problems (DKPs).

The bounds $\beta' \le \beta$ are also $\in \mathbb{Z}_{>0}$, and are chosen such that $P \cap \mathbb{Z}^n = \phi$, i.e., DKP is integer-infeasible.

In the example, $\bar{p} = \begin{bmatrix}1\\1\end{bmatrix}$, $M = 20$, $\bar{r} = \begin{bmatrix}1\\-1\end{bmatrix}$, $\bar{a} = \begin{bmatrix}21\\19\end{bmatrix}$, $\beta' = 106$, $\beta = 113$.

Further, if we consider $\max\{\bar{p}^T\bar{x} \mid \bar{x} \in P\}$ and $\min\{\bar{p}^T\bar{x} \mid \bar{x} \in P\}$, we get values of the form $(k+1) - \delta$ and $k + \delta'$ where $0 < \delta, \delta' < 1$. for some $k \in \mathbb{Z}$. Hence branching on $\bar{p}^T\bar{x}$ proves infeasibility at the root node.

How do we locate the "good" direction $\bar{p}$ (we do not know beforehand that $\bar{a}$ decomposes in this fashion)?

# Lattices

The lattice spanned by $[\bar{b}_1, \ldots, \bar{b}_n]$, $\bar{b}_i \in \mathbb{R}^m$, $i = 1, \ldots, n$, is the set of integer linear combinations of $\bar{b}_i$'s.

$$\mathcal{L}([\bar{b}_1, \ldots, \bar{b}_n]) = \left\{ \sum_{j=1}^{n} \bar{b}_j x_j \;\middle|\; x_j \in \mathbb{Z} \; \forall j \right\}.$$

With $B = [\bar{b}_1 \ldots \bar{b}_n]$, we can write

$$\mathcal{L}(B) = \{ B\bar{x} \mid \bar{x} \in \mathbb{Z}^n \}. \qquad B \in \mathbb{R}^{m \times n}$$

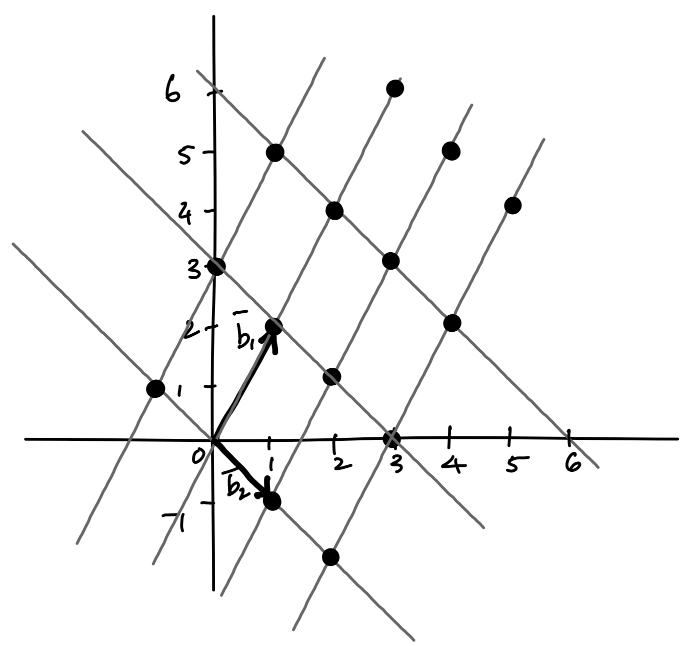
Note: $\text{span}(B) = \{ B\bar{y} \mid \bar{y} \in \mathbb{R}^n \}$ is the linear subspace spanned by the columns of $B$.

One could think of lattices as the main spaces in "integer linear algebra".

e.g., $\bar{b}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $\bar{b}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$.

$B = [\bar{b}_1 \; \bar{b}_2]$

$\mathcal{L}(B)$ consists of the grid points shown here as $\bullet$.

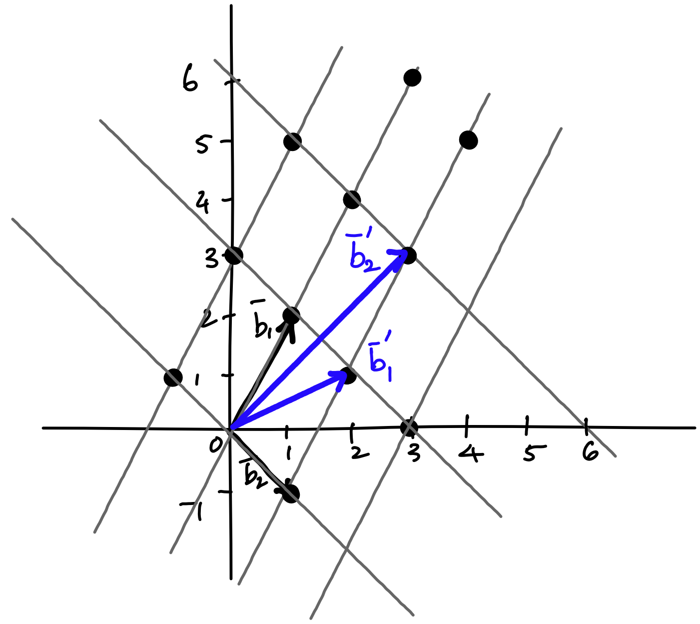


With $\bar{b}_1' = \bar{b}_1 + \bar{b}_2 = \begin{bmatrix}1\\2\end{bmatrix} + \begin{bmatrix}1\\-1\end{bmatrix} = \begin{bmatrix}2\\1\end{bmatrix}$ and $\bar{b}_2' = 2\bar{b}_1 + \bar{b}_2 = \begin{bmatrix}3\\3\end{bmatrix}$,
we get another basis $B' = [\bar{b}_1' \ \bar{b}_2']$ for $\mathcal{L}(B)$.

In the example, we had $\mathcal{L}(B) = \mathcal{L}(B')$

with $B = [\bar{b}_1 \ \bar{b}_2] = \begin{bmatrix}1 & 1\\2 & -1\end{bmatrix}$ and

$B' = [\bar{b}_1' \ \bar{b}_2'] = \begin{bmatrix}2 & 3\\1 & 3\end{bmatrix}$.




Note: $B' = BU$ where $U = \begin{bmatrix}1 & 2\\1 & 1\end{bmatrix}$.

$$\underbrace{\begin{bmatrix}2 & 3\\1 & 3\end{bmatrix}}_{B'} = \underbrace{\begin{bmatrix}1 & 1\\2 & -1\end{bmatrix}}_{B}\underbrace{\begin{bmatrix}1 & 2\\1 & 1\end{bmatrix}}_{U}.$$ Here

$\det(U) = -1$.

In general, given two bases $B, B'$ for a lattice $\mathcal{L}$, we have $B' = BU$ for unimodular $U$, i.e., $\det(U) = \pm 1$.

Any vector in $\mathcal{L}(B)$ is $\bar{v} = B\bar{x}$, $\bar{x} \in \mathbb{Z}^2$. We can write

$$\bar{v} = B\bar{x} = BUU^{-1}\bar{x} = B'\bar{x}'.$$

Notice that $U^{-1}$ has integer entries, as $\det U = \pm 1$.

Also, $\bar{x}$ and $\bar{x}'$ are unique for a given $\bar{v}$.

# MATH 567: Lecture 26 (04/15/2025)

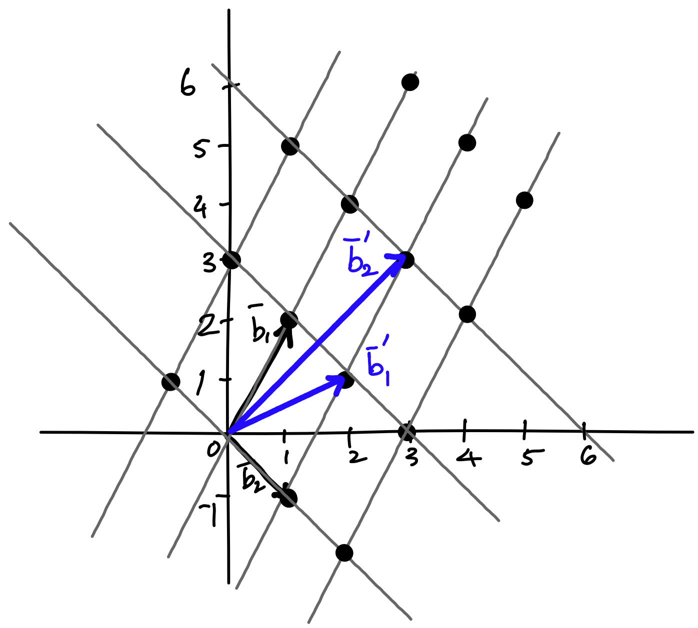
Today: * lattices and basis reduction

**Recall:** The lattice generated by $B \in \mathbb{R}^{m \times n}$ is $\mathcal{L}(B) = \{B\bar{x} \mid \bar{x} \in \mathbb{Z}^n\}$.
Here, $B$ is a basis for the lattice.

In the example, we had $\mathcal{L}(B) = \mathcal{L}(B')$ with

$$B = [\bar{b}_1 \; \bar{b}_2] = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \quad \text{and}$$

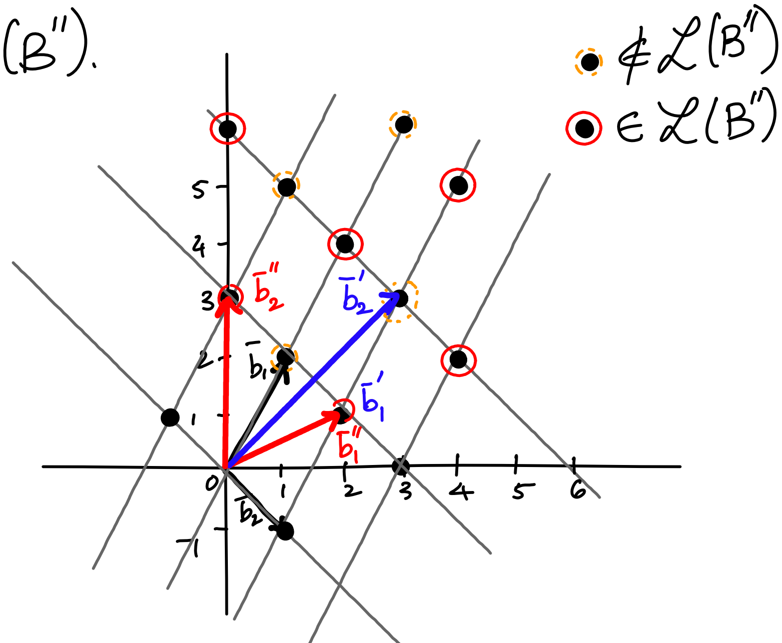
$$B' = [\bar{b}_1', \bar{b}_2'] = \begin{bmatrix} 2 & 3 \\ 1 & 3 \end{bmatrix}.$$




Note: $B' = BU$ where $U = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix}$.

$$\underbrace{\begin{bmatrix} 2 & 3 \\ 1 & 3 \end{bmatrix}}_{B'} = \underbrace{\begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix}}_{U}. \quad \det(U) = -1.$$

Now, consider $\bar{b}_1'' = \bar{b}_1 + \bar{b}_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ and $\bar{b}_2'' = \bar{b}_1 - \bar{b}_2 = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$.
With $B'' = [\bar{b}_1'' \; \bar{b}_2'']$, we see that $\mathcal{L}(B'') \subset \mathcal{L}(B)$,
as $\bar{b}_1 \in \mathcal{L}(B)$, but $\bar{b}_1 \notin \mathcal{L}(B'')$.




Note that $\bar{b}_1 = \frac{1}{2}(\bar{b}_1'' + \bar{b}_2'')$,
and hence we cannot express $\bar{b}_1$ as an integer linear combination of $\bar{b}_1''$ and $\bar{b}_2''$

$\mathcal{L}(B'')$ is a **sublattice** of $\mathcal{L}(B)$.

# Two fundamental problems in lattices

1. Shortest Vector Problem (SVP): Given a basis $B \in \mathbb{Z}^{m \times n}$ for lattice $\mathcal{L}(B)$, find $\bar{x} \in \mathbb{Z}^n / \{\bar{0}\}$ such that $\|B\bar{x}\| \leq \|B\bar{y}\| \; \forall \; \bar{y} \in \mathbb{Z}^n / \{\bar{0}\}$.

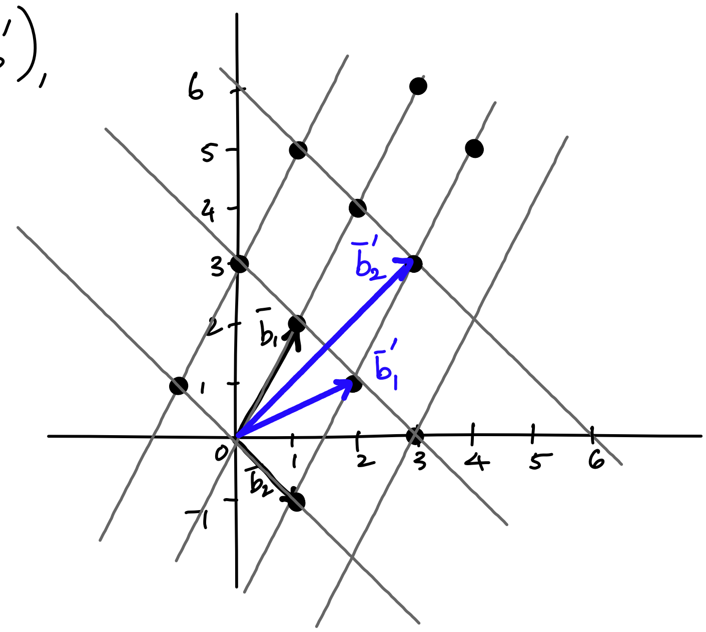
Euclidean norm

In words, find shortest nonzero vector in $\mathcal{L}(B)$.

2. Closest Vector Problem (CVP) Given a basis $B \in \mathbb{Z}^{m \times n}$ and a target vector $\bar{t} \in \mathbb{R}^m$, find $\bar{x} \in \mathbb{Z}^n$ such that $\|B\bar{x} - \bar{t}\| \leq \|B\bar{y} - \bar{t}\| \; \forall \; \bar{y} \in \mathbb{Z}^n$.

In words, find the closest vector in $\mathcal{L}(B)$ to $\bar{t}$ (could be $\bar{0}$).

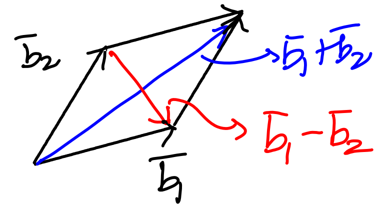
With $B = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix}$, $\bar{b}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ is a shortest vector of $\mathcal{L}(B)$. $\bar{b}_2$ is also a shortest vector of $\mathcal{L}(B')$, as $\mathcal{L}(B') = \mathcal{L}(B)$.




SVP in 2D can be solved in polynomial time by Gauss Reduction.

**Def** $B = [\bar{b}_1\ \bar{b}_2]$ with $\bar{b}_1, \bar{b}_2 \in \mathbb{Z}^2$ is **reduced** if

$$\|\bar{b}_1\|, \|\bar{b}_2\| \leq \|\bar{b}_1 + \bar{b}_2\|, \|\bar{b}_1 - \bar{b}_2\|.$$




The sides of the parallelogram are not longer than its diagonals.

$\bar{b}_1$ in a reduced basis will be a shortest vector of $\mathcal{L}(B)$.

We present the standard notion of orthogonalization in $\mathbb{R}^m$ — we will use it as a guide for reduction using integer multipliers.

## Gram-Schmidt Orthogonalization (GSO) (in $\mathbb{R}^m$)

$B = [\bar{b}_1, \ldots, \bar{b}_n]$, $\bar{b}_j \in \mathbb{R}^m\ \forall j$

$B^* = GSO(B)$

$\bar{b}_1^* = \bar{b}_1$;

for $i = 1, \ldots, n$

$$\bar{b}_i^* = \bar{b}_i - \sum_{j=1}^{i-1} \mu_{ij} \bar{b}_j^*$$

end

where $\mu_{ij} = \dfrac{\bar{b}_i^T \bar{b}_j^*}{\|\bar{b}_j^*\|^2} \left(\text{or } \dfrac{\langle \bar{b}_i, \bar{b}_j^* \rangle}{\|\bar{b}_j^*\|^2}\right)$ for $j < i$, and

$$\mu_{ii} = 1\ \forall i, \quad \mu_{ij} = 0\ \forall j > i.$$

$\mu_{ij}$ = length of component of $\bar{b}_i$ in direction of $\bar{b}_j^*$.

# Gauss Reduction (in 2D)

$[\tilde{\bar{b}}_1\ \tilde{\bar{b}}_2] = \text{GAUSS}(\bar{b}_1, \bar{b}_2);$ input : $\bar{b}_1, \bar{b}_2 \in \mathbb{Z}^2$

do
    if $\|\bar{b}_1\| > \|\bar{b}_2\|$
        swap $(\bar{b}_1, \bar{b}_2)$;
    end-if
    $\mu = \left\lfloor \frac{\langle \bar{b}_2, \bar{b}_1 \rangle}{\|\bar{b}_1\|^2} \right\rceil;$
    $\bar{b}_2 = \bar{b}_2 - \mu \bar{b}_1;$
    if $\|\bar{b}_1\| \leq \|\bar{b}_2\|$
        return $([\bar{b}_1, \bar{b}_2])$;
        break; $\longrightarrow$ this terminates the algorithm
    end-if
while $(\|\bar{b}_1\| > \|\bar{b}_2\|)$

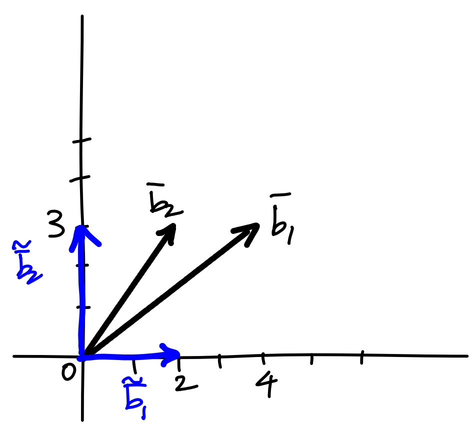
## Example

$$B = [\bar{b}_1\ \bar{b}_2] = \begin{array}{cc} \bar{b}_1 & \bar{b}_2 \end{array} \begin{bmatrix} 4 & 2 \\ 3 & 3 \end{bmatrix}$$

1. $\|\bar{b}_2\| < \|\bar{b}_1\| \Rightarrow$ swap $(\bar{b}_1, \bar{b}_2)$

$$\mu = \left\lfloor \frac{\langle \bar{b}_2, \bar{b}_1 \rangle}{\|\bar{b}_1\|^2} \right\rceil = \left\lfloor \frac{17}{13} \right\rceil = 1.$$

$$\bar{b}_2 = \bar{b}_2 - \mu \bar{b}_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}.$$




2. $B = \begin{array}{cc} \bar{b}_1 & \bar{b}_2 \end{array} \begin{bmatrix} 2 & 2 \\ 3 & 0 \end{bmatrix}.$

swap $\Longrightarrow$ $B = \begin{bmatrix} 2 & 2 \\ 0 & 3 \end{bmatrix}.$

$\mu = \left\lfloor \frac{4}{4} \right\rceil = 1.$ $\bar{b}_2 = \begin{bmatrix} 2 \\ 3 \end{bmatrix} - \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}.$

But, finding a shortest vector in $n \geq 3$ dimensions is hard!

$\tilde{B} = B = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}.$ $\tilde{\bar{b}}_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$ is a shortest vector in $\mathcal{L}(B)$.

We now define reduced bases in higher dimensions.

## Reduced bases for $n \geq 3$

We need some more notation.

Let $\bar{b}_i(\ell) = \sum_{j=\ell}^{n} \mu_{ij} \bar{b}_j^*$

$\bar{b}_i(\ell)$ is the component of $\bar{b}_i$ orthogonal to $\bar{b}_1^*, \bar{b}_2^*, \ldots, \bar{b}_{\ell-1}^*$.

Also, let $\mathcal{L}_i = \mathcal{L}([\bar{b}_i(i), \ldots, \bar{b}_n(i)])$.

e.g., $\bar{b}_{i+1}(i) = \mu_{i+1,i} \bar{b}_i^* + \bar{b}_{i+1}^*$, which is the component of $\bar{b}_{i+1} \perp \bar{b}_1^*, \bar{b}_2^*, \ldots, \bar{b}_{i-1}^*$.

## Korkine-Zolotarev (KZ) reduction

$B = [\bar{b}_1, \ldots, \bar{b}_n]$ is KZ-reduced if

* $\bar{b}_1$ is an SV of $\mathcal{L}(B)$; → shortest vector
* for $i \geq 2$
  $\bar{b}_i(i)$ is an SV of $\mathcal{L}_i$.

Notice that Gauss reduction $\equiv$ KZ-reduction in 2D.

Thus, KZ-reduction specifies quite a strong condition for a basis's being reduced, as the shortest vector conditions are imposed on larger and larger subsets of vectors (and not just on pairs of them).

If $B$ is KZ-reduced, then

$$\frac{4}{i+3} \leq \frac{\|\bar{b}_i\|^2}{\lambda_i^2(\mathcal{L})} \leq \frac{i+3}{4}, \text{ for } i=1,\dots,n$$

where $\lambda_i(\mathcal{L})$ = length of a shortest vector in $\mathcal{L}_i$.

Thus for $i=1$, $\|b_1\| = \lambda_1(\mathcal{L}) = \lambda(\mathcal{L})$, i.e., $\bar{b}_1$ is an SV of $\mathcal{L}(B)$.

For $i \geq 2$, $\|b_i\|$ is at most $\sqrt{n}$ off from $\lambda_i$, the $i^{th}$ minimum of the lattice.

While KZ-reduction is strong in its enforcement, computing a KZ-reduced basis starting from any basis is hard (no polynomial time algorithm is known). We consider less strict definitions that could be computed efficiently.

We first give an equivalent definition of Gauss reduction using the GSO coefficients $\mu_{ij}$. This definition can be more easily extended to higher dimensions.

Equivalent definition of Gauss reduction:

$B = [\bar{b}_1, \bar{b}_2]$ is Gauss-reduced if

$$\|\bar{b}_1\|^2 \leq \|\bar{b}_2\|^2$$

and $$\left| \frac{\langle \bar{b}_2, \bar{b}_1 \rangle}{\|\bar{b}_1\|^2} \right| \leq \frac{1}{2}.$$

We will round $\frac{1}{2}$ to 0. With this assumption, $\lfloor \mu \rceil = 0$.

# MATH 567: Lecture 27 (04/17/2025)

Today: * LLL reduction
* Hermite Normal Form (HNF)

## Lenstra-Lenstra-Lovász (LLL) Reduction (1982) (A.K. Lenstra, H.W. Lenstra)

Let $B = [\bar{b}_1, \dots, \bar{b}_n]$ and $B^* = [\bar{b}_1^*, \dots, \bar{b}_n^*] = GSO(B)$.

$B$ is **LLL-reduced** if

(1) $|\mu_{ij}| \leq \frac{1}{2}$ for $1 \leq j < i \leq n$, and

(2) $\|\bar{b}_i^*\|^2 \leq \frac{4}{3} \|\bar{b}_{i+1}^* + \mu_{i+1,i} \bar{b}_i^*\|^2$ for $i = 1, \dots, n-1$.

$\frac{4}{3} \longrightarrow$ can replace by $1+\varepsilon$ for any $\varepsilon > 0$.

Condition (1) says that $\bar{b}_i$'s are "nearly orthogonal". Recall that the GSO coefficient $\mu_{ij}$ gives the length of the component/projection of $\bar{b}_i$ along $\bar{b}_j^*$. Having an upper bound of $\frac{1}{2}$ on $|\mu_{ij}|$ specifies that these components are not too large.

Condition (2) says that the $\bar{b}_i$'s are "relatively short". With just Condition (1), we could have nearly orthogonal vectors, but $\|\bar{b}_1\|$ could be huge. And even though the $\bar{b}_i$ for $i \geq 2$ are "spread out" according to Condition (1) they could all have large norms! Notice that $\bar{b}_{i+1}^* + \mu_{i+1,i} \bar{b}_i^*$ is the component of $\bar{b}_{i+1}$ orthogonal to $\bar{b}_1^*, \dots, \bar{b}_{i-1}^*$, and $\bar{b}_i^*$ is the component of $\bar{b}_i$ orthogonal to $\bar{b}_1^*, \dots, \bar{b}_{i-1}^*$. The factor $\frac{4}{3}$ ensures the LLL-reduction runs in polynomial time — but it could be replaced by $1+\varepsilon$ for any $\varepsilon > 0$.

# Properties of LLL-reduced basis

Recall, $B = [\bar{b}_1, \dots, \bar{b}_n]$, $B^* = GSO(B)$.

(i) $\|\bar{b}_i^*\|^2 \leq 2^{j-i} \|\bar{b}_j^*\|^2 \;\; \forall \;\; 1 \leq i < j \leq n.$

(ii) $\|\bar{b}_1^*\| = \|\bar{b}_1\| \leq 2^{(n-1)/4} [\det(\mathcal{L})]^{1/n}$, where

$$\det(\mathcal{L}) = \prod_{j=1}^{n} \|\bar{b}_j^*\| \quad \text{(determinant of lattice } \mathcal{L})$$

When $m = n$, and $\bar{b}_j$ are rational, $\det(\mathcal{L}) = \sqrt{\det(B^T B)}$.

(iii) $\|\bar{b}_1^*\| = \|\bar{b}_1\| \leq 2^{(n-1)/2} \lambda(\mathcal{L})$. → length of a shortest vector in $\mathcal{L}$.

(iv) $\|\bar{b}_1\| \cdots \|\bar{b}_n\| \leq 2^{n(n-1)/4} \det(\mathcal{L})$.

## Proof of (i)

(i) $\|\bar{b}_i^*\|^2 \leq 2^{j-i} \|\bar{b}_j^*\|^2$, $j > i$.

Condition (2) of LLL-reduction $\Rightarrow$

$$\frac{3}{4} \|\bar{b}_i^*\|^2 \leq \|\bar{b}_{i+1}^* + \mu_{i+1,i} \bar{b}_i^*\|^2$$

$$\leq \|\bar{b}_{i+1}^*\|^2 + (\mu_{i+1,i})^2 \|\bar{b}_i^*\|^2$$

as $\|\bar{a} + \bar{b}\|^2 \leq \|\bar{a}\|^2 + \|\bar{b}\|^2 + \cancel{2\langle \bar{a}, \bar{b} \rangle}$ $= 0$, $\bar{b}_{i+1}^* \perp \bar{b}_i^*$

$$\leq \|\bar{b}_{i+1}^*\|^2 + \frac{1}{4} \|\bar{b}_i^*\|^2$$

$$\Rightarrow \|\bar{b}_i^*\|^2 \leq 2 \|\bar{b}_{i+1}^*\|^2 \quad \forall \; i = 1, \dots, n-1.$$

$$\Rightarrow \|\bar{b}_i^*\|^2 \leq \alpha \cdot \alpha \|\bar{b}_{i+2}^*\|^2 \leq \alpha^{j-i} \|\bar{b}_j^*\|^2, \forall j > i. \quad \square$$

Note (iii) $\|\bar{b}_1\| \leq \alpha^{(n-1)/2} \lambda(\mathcal{L})$

The length of $\bar{b}_1$ is at most an exponential factor off from the length of the SV of $\mathcal{L}$. But in practice $\|\bar{b}_1\|$ is often much closer to $\lambda(\mathcal{L})$ than the exponential bound seems to suggest.

Further, the LLL-algorithm runs in polynomial time. There are efficient implementations available as well — see, e.g., https://libntl.org/.

## Hermite Normal Form (HNF)

Let $P = \{\bar{x} \in \mathbb{R}^n \mid A\bar{x} = \bar{b}, \bar{x} \geq \bar{0}\}$, $A \in \mathbb{Z}^{m \times n}$, $\bar{b} \in \mathbb{Z}^m$.

Q: Is $P \cap \mathbb{Z}^n = \phi$?

This is the IP feasibility problem, and is NP-complete.

But if we remove $\bar{x} \geq \bar{0}$ from the definition of $P$, the problem is solvable in poly-time.

$\{\bar{x} \mid A\bar{x} = \bar{b}, \bar{x} \in \mathbb{Z}^n\}$: system of linear Diophantine equations.

**Def** Let $A \in \mathbb{Z}^{m \times n}$ with $\text{rank}(A) = m$. $A$ is in Hermite normal form (HNF) if $A = [B \; 0]$ where $B \in \mathbb{Z}^{m \times m}$ is

1. non-singular $(\det(B) \neq 0)$,
2. non-negative,
3. lower-triangular, and
4. every row of $B$ has a unique maximum entry located on the main diagonal, i.e., $B_{ii} > B_{ij} \; \forall j$.

$$A = \begin{bmatrix} * & 0 & & \\ * & * & & 0 \\ * & \ddots & * & \\ & & * & * \end{bmatrix}$$

$*$ = (can be) non-zero

largest in each row, $>0$.

e.g., $A = \begin{bmatrix} 2 & 0 & 0 \\ 1 & 3 & 0 \end{bmatrix}$ is in HNF.

$B$

$A$ can be converted to HNF using elementary column operations (ECOs). With $A = [\bar{a}_1 \cdots \bar{a}_n] \in \mathbb{Z}^{m \times n}$, the ECOs are

1. $\bar{a}_i \rightleftarrows \bar{a}_j$ (swap two columns),
2. $\bar{a}_i \leftarrow -\bar{a}_i$ (scale column by $-1$), and
3. $\bar{a}_i \leftarrow \bar{a}_i + \lambda \bar{a}_j, \; \lambda \in \mathbb{Z}$ (add integer multiple of column $j$ to column $i$)

$$\text{HNF}([5 \; 2]) = [1 \; 0]$$
$$\text{HNF}([6 \; 2]) = [2 \; 0]$$

If $\alpha_i \in \mathbb{Z}$, then $\text{HNF}([\alpha_1 \, \alpha_2 \cdots \alpha_n]) = [\gcd(\alpha_1, \ldots, \alpha_n) \; 0 \cdots 0]$.

## Example

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 10 \end{bmatrix} \xrightarrow[C_3 - 3C_1]{C_2 - 2C_1} \begin{bmatrix} 1 & 0 & 0 \\ 4 & -3 & -6 \\ 7 & -6 & -11 \end{bmatrix} \xrightarrow[\text{then } C_3 + 2C_2]{-C_2} \begin{bmatrix} 1 & 0 & 0 \\ 4 & 3 & 0 \\ 7 & 6 & 1 \end{bmatrix}$$

$$\xrightarrow{C_1 - C_2} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 3 & 0 \\ 1 & 6 & 1 \end{bmatrix} \xrightarrow[C_2 - 6C_3]{C_1 - C_3} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

is in HNF!

**Theorem 18** (Theorem 4.1 from Schrijver TLIP): $A$ can be brought into HNF using ECO's. The numbers stay bounded in the process.

**Proof (of first statement)** Suppose (after some steps)

$$A = \begin{array}{c} \\ k \\ \\ \end{array} \begin{array}{c} \overset{k}{\overbrace{\qquad}} \\ \left[\begin{array}{c|c} B & 0 \\ \hline C & D \end{array}\right] \end{array}$$

with $B_{k \times k}$ lower triangular, and $\mathrm{diag}(B) > 0$.

$[d_{11}\ d_{12} \cdots d_{1\ell}]$, $\ell = n-k$

We can use ECOs to ensure that

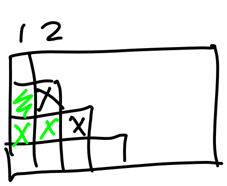
(i) the first row of $D = [d_{11} \cdots d_{1\ell}]$ is $\geq 0$, and

(ii) $d_{11} + d_{12} + \cdots + d_{1\ell}$ is minimal.

In fact, we can get $d_{11} > 0$, and $d_{1j} = 0$, $j = 2, \ldots, \ell$.

Hence we have increased the size of $B$ to $(k+1) \times (k+1)$.

Can use further ECOs to get diagonal dominance property.




Use column 2 to make (2,1)-entry "okay".

Then use column 3 to make Row-3 "okay"!

And so on... □

# "Farkas' lemma" for IP

$\{A\bar{x} \le \bar{b}\}$ is infeasible $\Rightarrow \exists\, \bar{u} \ge \bar{0}$ s.t. $\bar{u}^T A = \bar{0}$, $\bar{u}^T \bar{b} = -1$.

As an application of HNF, we present a Farkas' lemma-type systems of alternatives result for IP.

(1) $\{A\bar{x} = \bar{b}, \bar{x} \in \mathbb{Z}^n\}$ has no solution.
$(A \in \mathbb{Z}^{m \times n},\ \bar{b} \in \mathbb{Z}^m)$

(2) $\exists\, \bar{y}$ rational such that $\bar{y}^T A$ is integral, $\bar{y}^T \bar{b}$ is non-integral.

"Farkas' lemma" for IP: $(1) \equiv (2)$.

**Proof** $(2) \Rightarrow (1)$: $\bar{y}^T(A\bar{x} = \bar{b}) \Rightarrow \underbrace{(\bar{y}^T A)}_{\in \mathbb{Z}^n}\bar{x} = \underbrace{\bar{y}^T \bar{b}}_{\notin \mathbb{Z}} \Rightarrow \bar{x} \notin \mathbb{Z}^n$.

$(1) \Rightarrow (2)$: We use HNF(A):

Note that (1) and (2) are both invariant under ECOs. Hence we can assume WLOG that $A$ is in HNF. With $HNF(A) = [B\ 0]$, where $B$ is non-singular, we get the following result.

$$B^{-1}[\underbrace{B\ \ 0}_{A}] = [I\ \ 0], \text{ i.e., } B^{-1}A \text{ is integral.}$$

$$B^{-1}(A\bar{x} = \bar{b}) \Rightarrow \underbrace{(B^{-1}A)}_{\in \mathbb{Z}^{m\times n}}\bar{x} = B^{-1}\bar{b}.$$

(1) $\Rightarrow B^{-1}\bar{b} \notin \mathbb{Z}^m$, i.e., with $\bar{u} = B^{-1}\bar{b}$, there is at least one $i$ such that $u_i \notin \mathbb{Z}$. We can choose $\bar{y}^T$ as the $i^{th}$ row of $B^{-1}$, and we get (2). $\square$

Note: HNF(A) can be computed in polynomial time, without any $A_{ij}$ becoming too big.