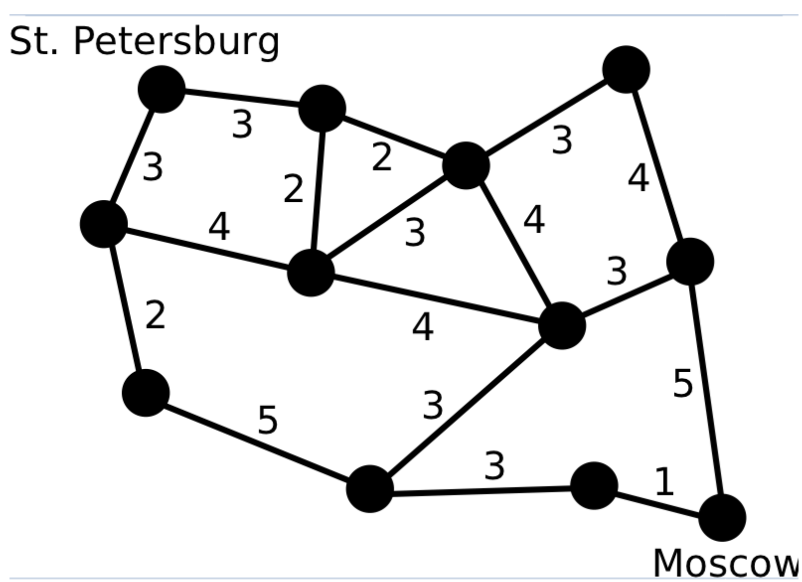
Introduction to Linear Programming

(Following "Understanding and using linear programming by Matoušek and Gärtner)

Etymology of linear programming:

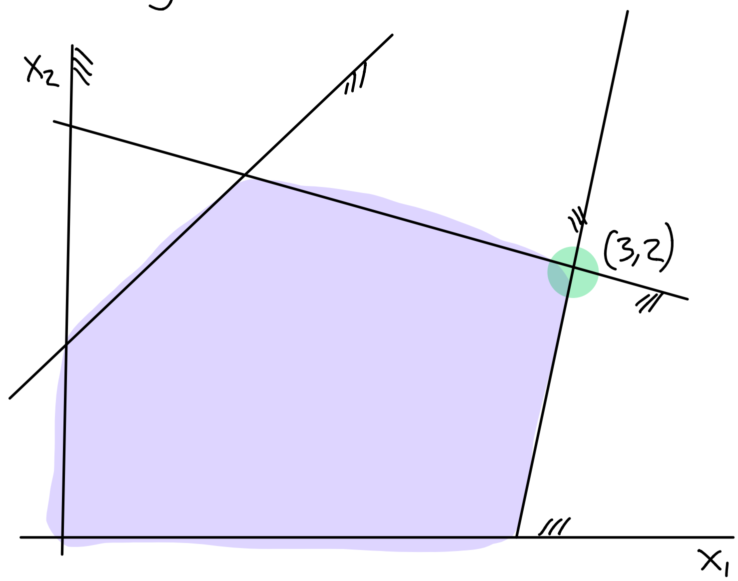
- 1950's, programming is military (not computer science) term: schedules, supply, deployment
- Better name: "planning with linear constraints"

Ex Maximal possible flow of concrete from St. Petersburg to Moscow?



<u>Ex</u> Maximize $x_1 + x_2$
for $x_1, x_2 \in \mathbb{R}$ satisfying

$$\begin{aligned} x_1 &\geq 0 \\ x_2 &\geq 0 \\ -x_1 + x_2 &\leq 1 \\ x_1 + 6x_2 &\leq 15 \\ 4x_1 - x_2 &\leq 10 \end{aligned}$$



<u>Vocabulary</u> Objective function, constraints, feasible solutions, optimal solution

<u>More generally</u>,

Maximize $c^T x$
subject to $Ax \leq b$

Here $x \in \mathbb{R}^n$ encodes the variables,
and $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$ are given.

Q What if we change c to $(\frac{1}{6}, 1)$?

Q What if we change to

$$-x_1 + x_2 \geq 1 \quad \text{and}$$
$$4x_1 - x_2 \geq 10 \quad ?$$

Infeasible

Q What if we remove the last two constraints?

Unbounded

Main point of class

Linear programs are efficiently solvable both

(i) in practice (good software, thousands of variables and constraints)

(ii) and in theory (algorithms bounded in time by polynomial functions of inputs).

Tension! Algorithms good for (i) are not the same as those for (ii).

Significance and History

Simplex algorithm chosen as one of top 10 algorithms with greatest significance on 20^{th} century

- Simplex algorithm, 1947, George Dantzig, US Air Force
- Leonid Kantarovich, USSR, 1939, timber industry organization
- 1975 Nobel Prize in Economics to Kantarovich and Tjalling Koopmans

- Entrepreneurs surprised when costs suddenly cut 20%
- Feasibility on large problems due as much to advancements in algorithms as to advancements in computer hardware.
- Big theoretical advancement: duality theorem
- Applications to convex and non-linear optimization.

<u>Analogy</u> Linear programming is like linear algebra over $\mathbb{R}^n_{\geq 0}$

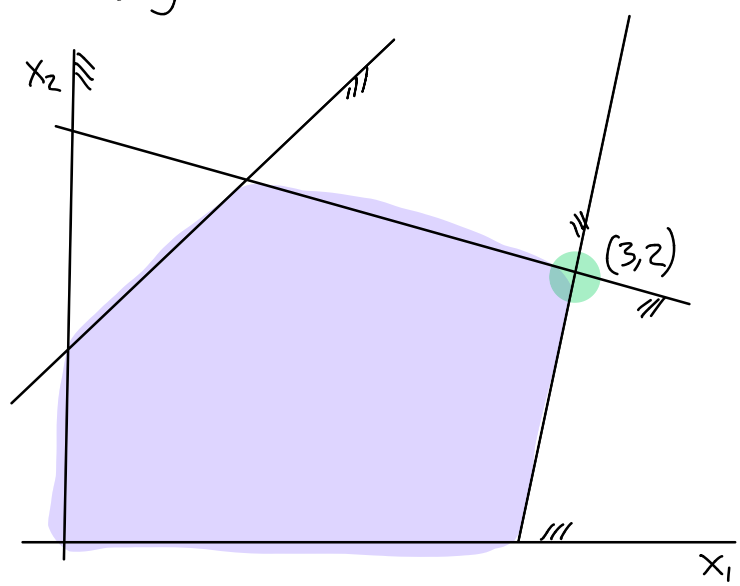
	Basic problem	Algorithm	Solution set
Linear algebra	Linear equations: $Ax = b$	Gaussian elimination	Affine subspace
Linear programming	Linear inequalities: $Ax \leq b$	Simplex method	Convex polyhedron

Indeed, in some sense finding an optimal solution is no harder than finding a feasible solution.

$\underline{\text{Ex}}$ Maximize $x_1 + x_2$

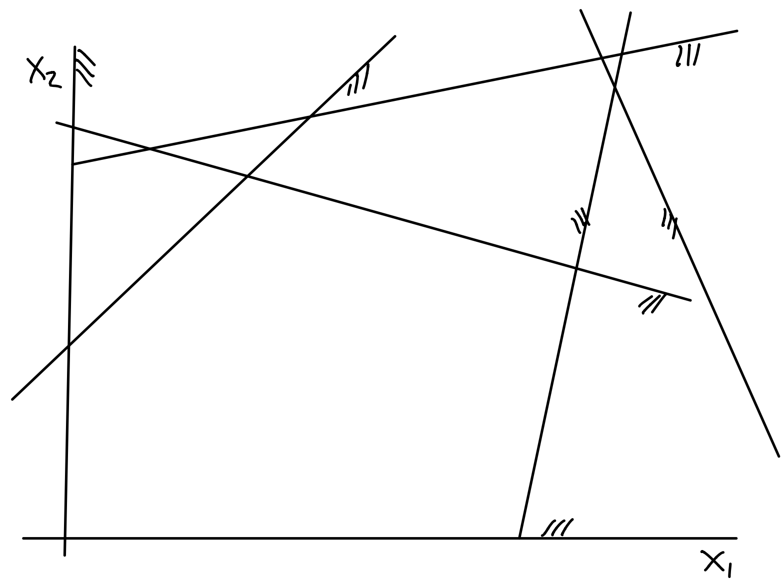
for $x_1, x_2 \in \mathbb{R}$ satisfying

$$x_1 \geq 0$$
$$x_2 \geq 0$$
$$-x_1 + x_2 \leq 1$$
$$x_1 + 6x_2 \leq 15$$
$$4x_1 - x_2 \leq 10$$



Polytopes, cubes, and cross-polytopes

$$\begin{aligned} x_1 &\geq 0 \\ x_2 &\geq 0 \\ -x_1 + x_2 &\leq 1 \\ x_1 + 6x_2 &\leq 15 \\ 4x_1 - x_2 &\leq 10 \\ &\vdots \end{aligned}$$



<u>Def</u> A <u>convex polyhedron</u> is an intersection of finetely many closed half-spaces in $\mathbb{R}^n$.

<u>Def 1</u> A <u>convex polytope</u> is a bounded convex polyhedron.

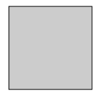
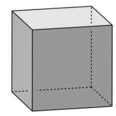
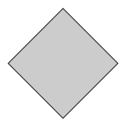
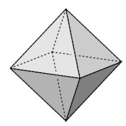
<u>Def 2</u> A <u>convex polytope</u> is the convex hull of a finite set of points.

<u>Thm (Minkowski–Weyl)</u> Def 1 and Def 2 agree.

Given a convex polytope described as half-spaces, it's not always easy to find the vertices!

Given a convex polytope described via vertices, it's not always easy to find the half-spaces!

Cubes and cross-polytopes

	$n=1$	$n=2$	$n=3$	$n=4$	n
Cube					
vertices	2	4	8	16	2^n
$(n-1)$-faces	2	4	6	8	$2n$
Cross-polytope					
vertices	2	4	6	8	$2n$
$(n-1)$-faces	2	4	8	16	2^n

The cube has exponentially many vertices compared to faces.
Hard in general to go from faces to vertices.

The cross-polytope has exponentially many faces compared to vertices.
Hard in general to go from vertices to faces.

They're dual to one-another!

Example linear programming application: Healthy diet

Food	Carrot, Raw (x_1)	White Cabbage, Raw (x_2)	Cucumber, Pickled (x_3)	Required per dish
Vitamin A [mg/kg]	35	0.5	0.5	0.5 mg
Vitamin C [mg/kg]	60	300	10	15 mg
Dietary Fiber [g/kg]	30	20	10	4 g
price [€/kg]	0.75	0.5	0.15*	—

Minimize $.75x_1 + .5x_2 + .15x_3$ (Maximize $-.75x_1 - .5x_2 + .15x_3$)

subject to

$x_1 \geq 0$

$x_2 \geq 0$

$x_3 \geq 0$

$35x_1 + .5x_2 + .5x_3 \geq .5$

$60x_1 + 300x_2 + 10x_3 \geq 10$

$30x_1 + 20x_2 + 10x_3 \geq 4$

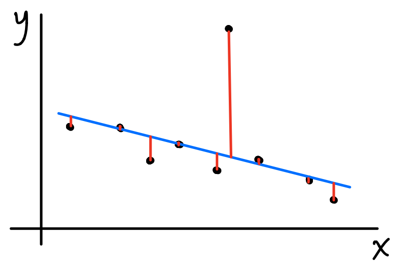
Optimal solution: $x_1 = 9.5g$ $x_2 = 38g$ $x_3 = 290g$

Cost = € 0.07

Dantzig with his first electronic computer access: Several litres vinegar, then 200 bouillon cubes.

Moral: Modeling is hard!

Example application: Fitting a line



$$y = ax + b$$

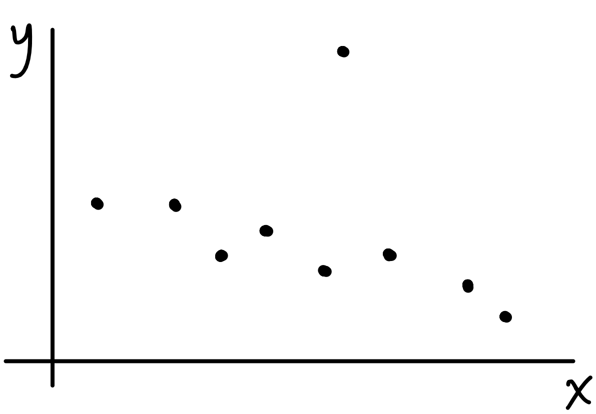
L^2 error Minimize $\sum_{i=1}^{n} (ax_i + b - y_i)^2$ [Susceptible to outliers]

Solve via linear regression

L^1 error Minimize $\sum_{i=1}^{n} |ax_i + b - y_i|$

Brainstorming:

Minimize $e_1 + e_2 + \dots + e_n$
subject to $e_i = |ax_i + b - y_i|$ for all i



Linear programming: Minimize $e_1 + e_2 + \dots + e_n$

subject to $e_i \geq ax_i + b - y_i$

$e_i \geq -(ax_i + b - y_i)$

for all i

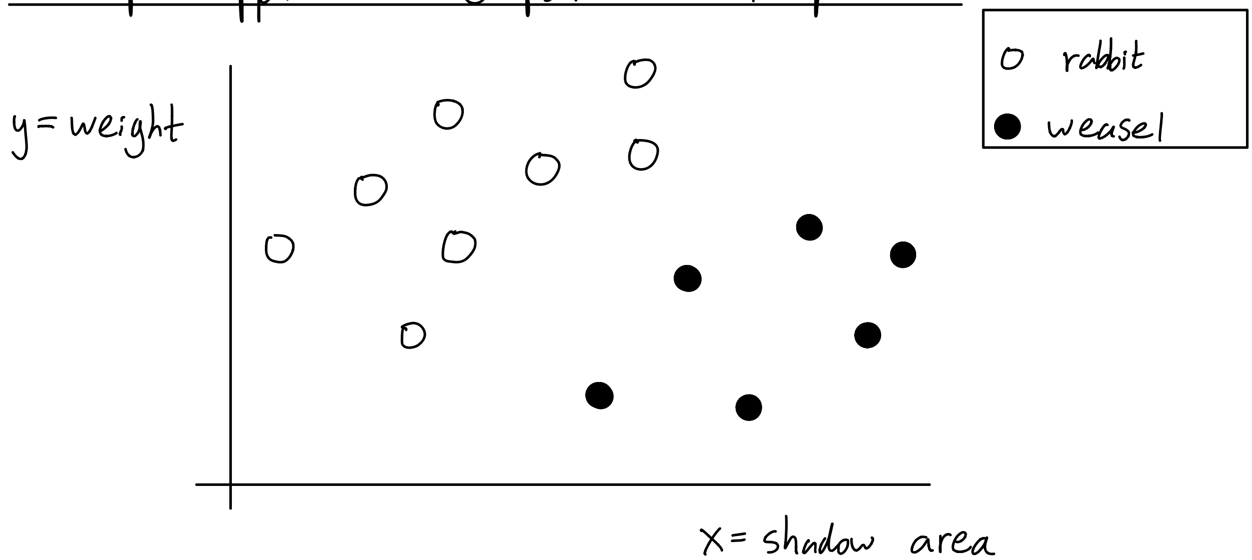
The constraints guarantee

$e_i \geq \max(ax_i + b - y_i, \; -(ax_i + b - y_i)) = |ax_i + b - y_i|$.

In an optimal solution, $\geq$ is satisfied with equality.

Moral: Objective functions or constraints with absolute values can often be handled by introducing extra variables or constraints.

Example application: Separation of points



Does there exist a separating line $y = ax + b$?

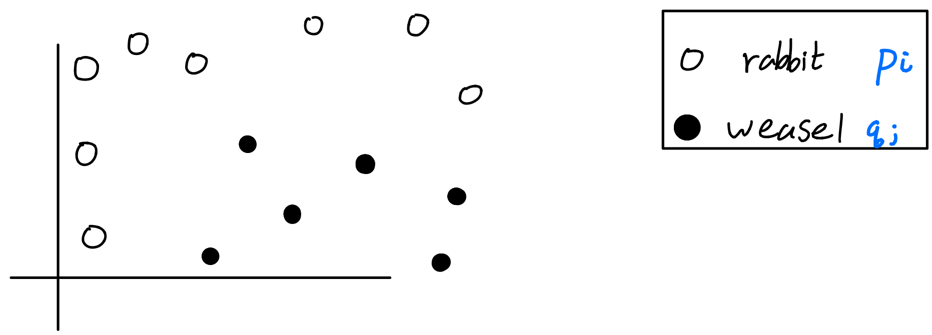
Case p_i points "above" q_j points:

Maximize δ

subject to $y(p_i) \geq a x(p_i) + b + \delta$ for all i

$y(q_j) \leq a x(q_j) + b - \delta$ for all j

Does there exist a separating parabola $y = ax^2 + bx + c$?



Maximize δ

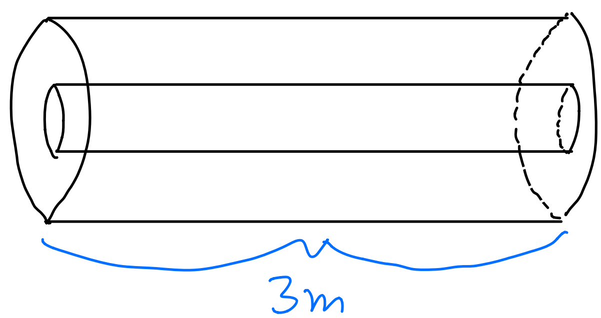
subject to $y(p_i) \geq a x(p_i)^2 + b x(p_i) + c + \delta$ for all i

$y(q_j) \leq a x(q_j)^2 + b x(q_j) + c - \delta$ for all j

Moral: Nonlinear functions can be incorporated into linear programming sometimes.

Example application: Cutting paper rolls

Paper mill makes 3 meter paper rolls.



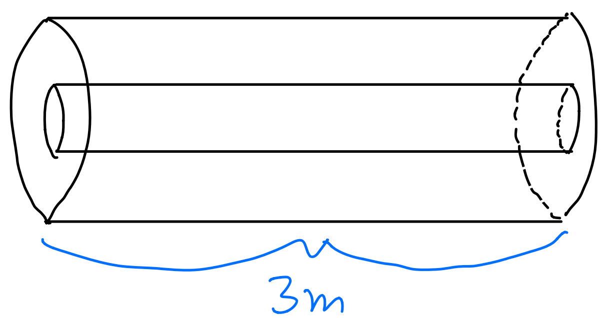
What's the fewest number of rolls need to satisfy an order of:

97	rolls	width	135 cm
610	rolls	width	108 cm
395	rolls	width	93 cm
211	rolls	width	42 cm

?

Example application: Cutting paper rolls

Paper mill makes 3 meter paper rolls.



What's the fewest number of rolls need to satisfy an order of:

97 rolls width 135 cm
610 rolls width 108 cm
395 rolls width 93 cm
211 rolls width 42 cm ?

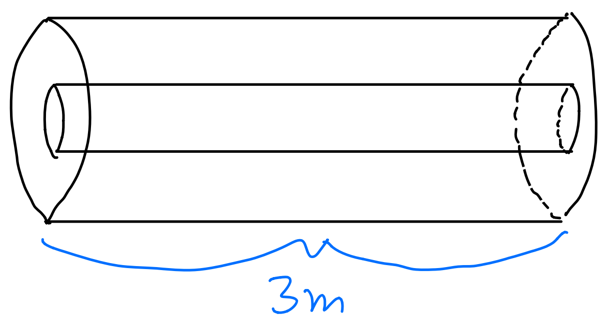
Possible ways to cut roll with <42cm wasted:

P1:
P2:
P3:
P4:
P5:
P6:
P7:
P8:
P9:
P10:
P11:
P12:

Generated by

Example application: Cutting paper rolls

Paper mill makes 3 meter paper rolls.



What's the fewest number of rolls need to satisfy an order of:

97 rolls width 135 cm
610 rolls width 108 cm
395 rolls width 93 cm
211 rolls width 42 cm ?

Possible ways to cut roll with <42cm wasted:

P1: $2 \cdot 135$
P2: $135 + 108 + 42$
P3: $135 + 93 + 42$
P4: $135 + 3 \cdot 42$
P5: $2 \cdot 108 + 2 \cdot 42$
P6: $108 + 2 \cdot 93$

Generated by computer

P7: $108 + 93 + 2 \cdot 42$
P8: $108 + 4 \cdot 42$
P9: $3 \cdot 93$
P10: $2 \cdot 93 + 2 \cdot 42$
P11: $93 + 4 \cdot 42$
P12: $7 \cdot 42$

For each possibility P_j, add a variable $x_j \geq 0$ representing # rolls cut that way.

Minimize $\sum_{j=1}^{12} x_j$ (total # of rolls cut)

subject to

For each possibility P_j, add a variable $x_j \geq 0$ representing # rolls cut that way.

Minimize $\sum_{j=1}^{12} x_j$ (total # of rolls cut)

subject to

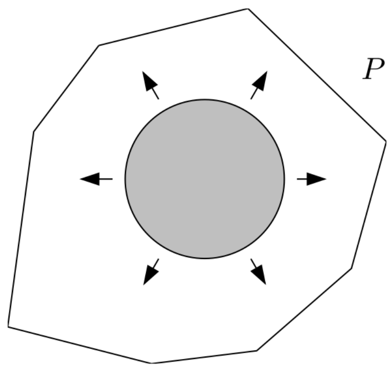
$$2x_1 + x_2 + x_3 + x_4 \geq 97$$
$$x_2 + 2x_5 + x_6 + x_7 + x_8 \geq 610$$
$$x_3 + 2x_6 + x_7 + 3x_9 + 2x_{10} + x_{11} \geq 395$$
$$x_2 + x_3 + 3x_4 + 2x_5 + 2x_7 + 4x_8 + 2x_{10} + 4x_{11} + 7x_{12} \geq 211$$

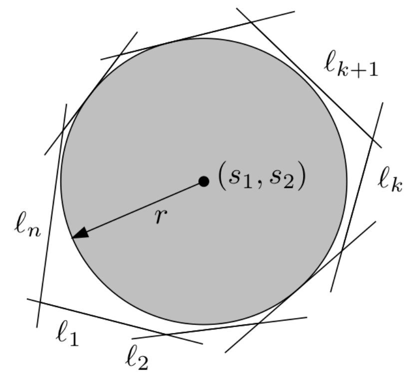
Optimal solution

$x_1 = 48.5$, $x_5 = 206.25$, $x_6 = 197.5$, all others zero.

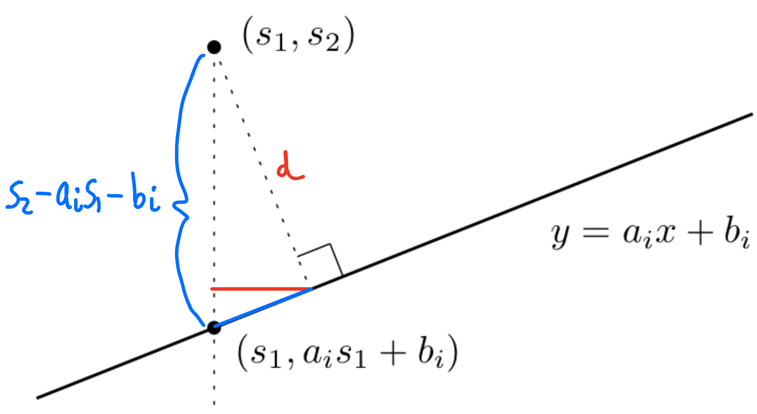
What if we want an integer solution?

Example Application: Largest disk in a polygon





The distance from the center point (s_1, s_2) to the line l_i $(y = a_i x + b_i)$ is the absolute value of $\dfrac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2 + 1}}$.



$$\frac{d}{s_2 - a_i s_1 - b_i} = \frac{1}{\sqrt{a_i^2 + 1}} \quad \text{so} \quad d = \frac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2 + 1}}.$$

We get the following linear program:

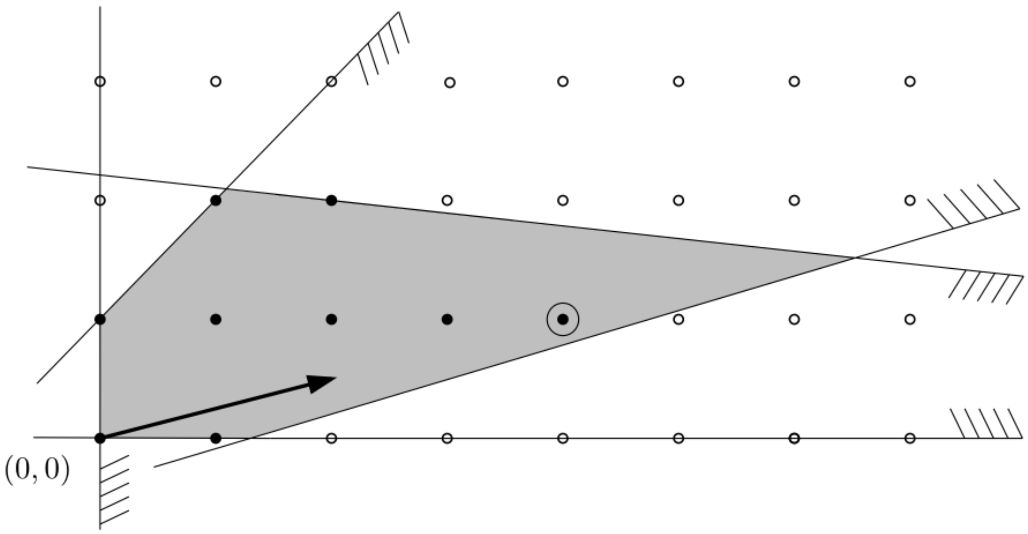
$$
\begin{aligned}
&\text{Maximize} \quad r \\
&\text{subject to} \quad \frac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2+1}} \geq \ \ r \quad \text{for } i = 1, 2, \ldots, k \\
&\qquad\qquad\quad\ \frac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2+1}} \leq -r \quad \text{for } i = k+1, k+2, \ldots, n.
\end{aligned}
$$

There are three variables: s_1, s_2, and r. An optimal solution yields the desired largest disk contained in P.

Can do in $\mathbb{R}^n$ instead of $\mathbb{R}^2$.

Finding the smallest disk containing a polygon is not linear but is convex (see Section 8.7).

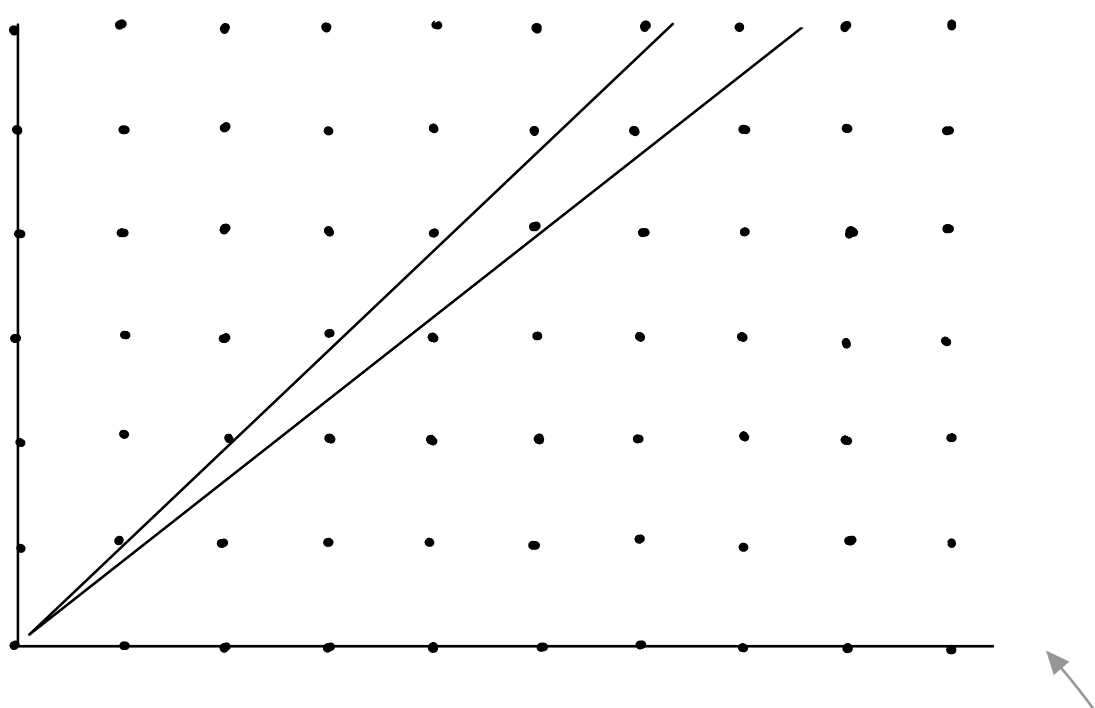
Integer Linear Programming



Maximize $c^T x$

subject to $Ax \leq b$

$x \in \mathbb{Z}^n$



In general, integer linear programming is <u>NP-hard</u>.

(10 variables, 10 constraints can be insurmountable.)

Techniques: cutting planes, branch and bound, branch and cut.

Is the most widespread use of linear programming today for integer programs?

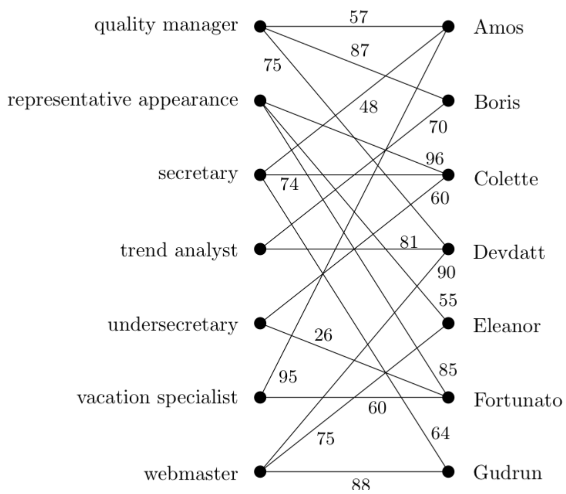
Mixed integer programming

Next: Example easy, medium, hard integer programs.

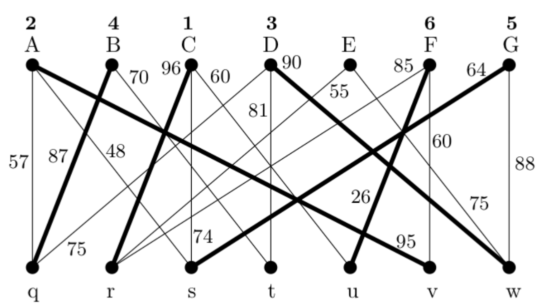
- Maximum weight matching
- Minimum vertex cover
- Maximum independent set

Maximum weight matching

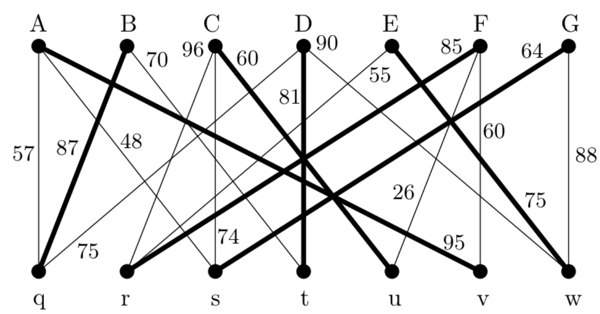
$G = (V, E)$



Greedy approach



Optimal solution



Integer program

maximize $\sum_{e \in E} w_e x_e$
subject to $\sum_{e \in E: v \in e} x_e = 1$ for each vertex $v \in V$, and
$x_e \in \{0, 1\}$ for each edge $e \in E$.

LP relaxation

Maximize $\sum_{e \in E} w_e x_e$
subject to $\sum_{e \in E: v \in e} x_e = 1$ for each vertex $v \in V$, and
$0 \le x_e \le 1$ for each edge $e \in E$.

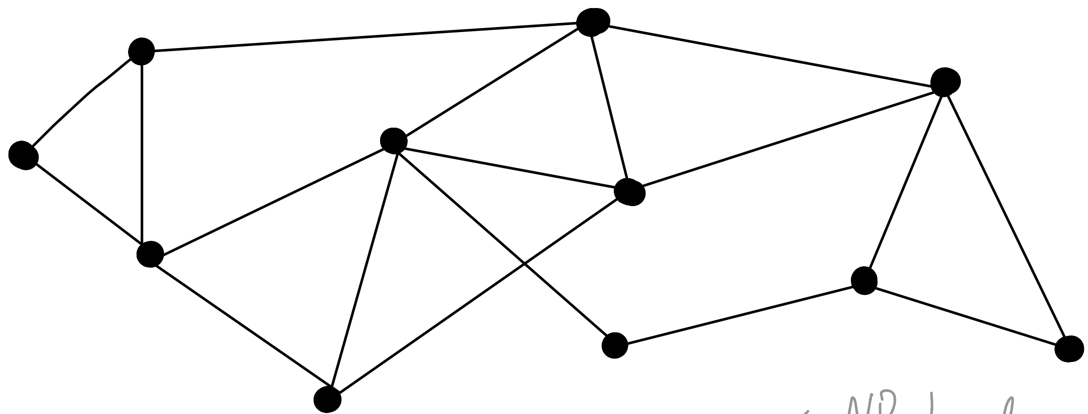
An optimal solution to any LP relaxation is a bound on an optimal solution to the integer program.

Thm 3.2.1 Let G be an arbitrary weighted bipartite graph. The LP relaxation above has an integer optimal solution (which also solves the integer program).

Moral Sometimes solving an integer program is no harder than solving a linear program.

Minimum vertex cover

$G=(V,E)$



NP-hard

IP:

$$\begin{array}{lll} \text{Minimize} & \sum_{v\in V} x_v & \\ \text{subject to} & x_u + x_v \geq 1 & \text{for every edge } \{u,v\}\in E \\ & x_v \in \{0,1\} & \text{for all } v\in V. \end{array}$$

LP relaxation $\quad 0 \leq x_v \leq 1$

Let $S_{IP} \subseteq V$ be a vertex cover solving IP.
Let $S_{LP} \subseteq V$ be a vertex cover solving LP, obtained by $S_{LP} = \{v\in V \mid x_v \geq \frac{1}{2}\}$.

Fact $\quad |S_{IP}| \leq |S_{LP}| \leq 2|S_{IP}|$

Proof of second inequality

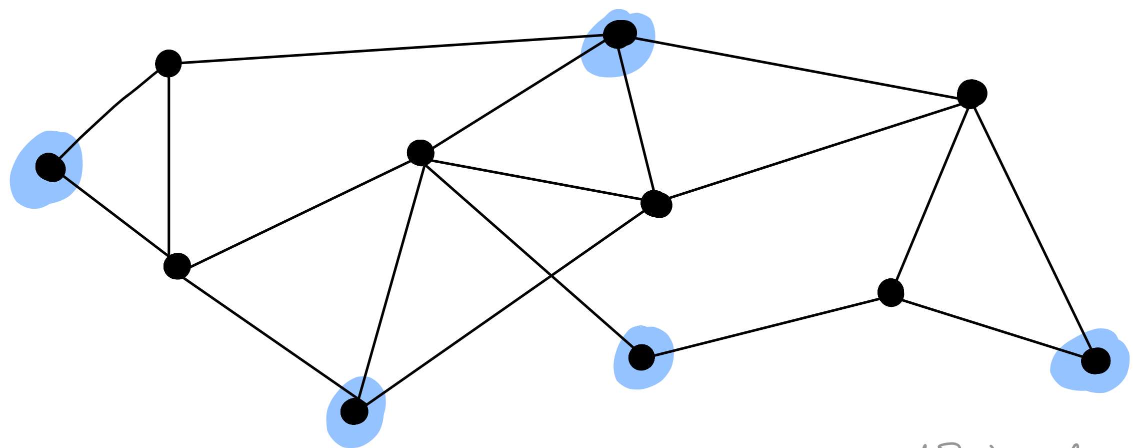
$$|S_{LP}| \leq 2\sum_{v\in V} x_v \leq 2\sum_{v\in V} \tilde{x}_v = 2\cdot|S_{IP}|$$

by definition of S_{LP}

Since any solution to IP is also a feasible solution for the LP relaxation.

Moral Sometimes solving an LP relaxation gives an approximate solution to an NP-hard integer program.

Maximum independent set $\quad G=(V,E)$



NP hard

IP:

Maximize	$\sum_{v \in V} x_v$	
subject to	$x_u + x_v \le 1$	for each edge $\{u,v\} \in E$, and
	$x_v \in \{0,1\}$	for all $v \in V$.

LP relaxation $\qquad 0 \le x_v \le 1$

In the LP relaxation $(0 \le x_v \le 1)$ we always have the feasible solution $x_v = \frac{1}{2}$ for all v, meaning the optimum is at least $\frac{1}{2}|V|$.

If G is the complete graph, then the IP has optimum 1.

__Moral__ Sometimes an LP relaxation tells us next to nothing about the integer program.

Approximation for this problem is known to be hard:

J. Håstad: Clique is hard to approximate within $n^{1-\varepsilon}$, *Acta Mathematica* 182(1999) 105–142,

$n = |V|$

Equational Form of a Linear Program

Any linear program can be rewritten as

$$\left.\begin{array}{ll} \text{Maximize} & c^T x \\ \text{subject to} & Ax \le b \end{array}\right\} \text{Not equational form}$$

with $c \in \mathbb{R}^n$, $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$.

Here $n =$ # variables and $m =$ # constraints

<u>Ex</u>

$$\begin{array}{ll} \text{Minimize} & 3x_1 + 4x_2 \\ \text{subject to} & 2x_1 - x_2 \ge 2 \\ & x_1 + x_2 = 3 \end{array}$$

becomes

$$\begin{array}{ll} \text{Maximize} & -3x_1 - 4x_2 \\ \text{subject to} & -2x_1 + x_2 \le -2 \\ & x_1 + x_2 \le 3 \\ & -x_1 - x_2 \le -3 \end{array}$$

with $c = \begin{bmatrix} -3 \\ -4 \end{bmatrix}$, $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $A = \begin{bmatrix} -2 & 1 \\ 1 & 1 \\ -1 & -1 \end{bmatrix}$, $b = \begin{bmatrix} -2 \\ 3 \\ -3 \end{bmatrix}$,
$n = 2$, $m = 3$.

The simplex method requires a different form, called *standard* or *equational* form:

$$\left.\begin{array}{ll} \text{Maximize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0 \end{array}\right\} \text{Equational form}$$

Transformation of an arbitrary linear program to equational form

Ex

$$\begin{array}{ll} \text{Maximize} & 3x_1 - 2x_2 \\ \text{subject to} & 2x_1 - x_2 \leq 4 \\ & x_1 + 3x_2 \geq 5 \\ & x_2 \geq 0 \end{array}$$

(1) $2x_1 - x_2 \leq 4$ becomes $2x_1 - x_2 + x_3 = 4$

$x_3 \geq 0$

slack variable

(2) $x_1 + 3x_2 \geq 5$ becomes $-x_1 - 3x_2 \leq -5$

and then $-x_1 - 3x_2 + x_4 = -5$

$x_4 \geq 0$

slack variable

(3) To handle the "missing" nonnegativity constraint, let $x_1 = x_1' - x_1''$ with $x_1' \geq 0$, $x_1'' \geq 0$.

Result: Maximize $3x_1' - 3x_1'' - 2x_2$

subject to

$$2x_1' - 2x_1'' - x_2 + x_3 = 4$$
$$-x_1' + x_1'' - 3x_2 + x_4 = -5$$
$$x_1' \geq 0,\ x_1'' \geq 0,\ x_2 \geq 0,\ x_3 \geq 0,\ x_4 \geq 0$$

(4) Then relabel $x_1', x_1'', x_2, x_3, x_4$

as x_1, x_2, x_3, x_4, x_5

Result: Maximize $3x_1 - 3x_2 - 2x_3$

subject to

$$2x_1 - 2x_2 - x_3 + x_4 = 4$$
$$-x_1 + x_2 - 3x_3 + x_5 = -5$$
$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

Here $c = \begin{bmatrix} 3 \\ -3 \\ -2 \end{bmatrix}$, $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$, $A = \begin{bmatrix} 2 & -2 & -1 & 1 & 0 \\ -1 & 1 & -3 & 0 & 1 \end{bmatrix}$, $b = \begin{bmatrix} 4 \\ -5 \end{bmatrix}$,

$n = 5$, $m = 2$.

$$
\begin{aligned}
\text{Maximize } \quad & c^T x \\
\text{subject to } \quad & Ax = b \\
& x \geq 0
\end{aligned}
\Bigg\} \text{ Equational form}
$$

<u>Remark</u> This translation takes us from n variables and m constraints ($\leq$, $\geq$, or $=$) to:

- at most $m+2n$ variables
- m equations
- all nonnegativity constraints

Basic feasible solutions — Algebra

We consider only linear programs in equational form

$$\begin{array}{ll} \text{Maximize} & c^T x \\ \text{subject to} & Ax = b \qquad (A \text{ size } m \times n) \\ & x \geq 0 \end{array}$$

such that

- $Ax = b$ has at least one solution, and
- the rows of A are linearly independent.

else we can find and delete rows from A

Else it is easy to determine the program is infeasible

<u>Def</u> A feasible solution $x \in \mathbb{R}^n$ is <u>basic</u> if there is an m-element set $B \subseteq \{1, 2, \ldots, n\}$ such that

- the square matrix A_B is nonsingular, i.e., the columns indexed by B are independent, and
- $x_j = 0$ for all $j \notin B$.

<u>Ex</u> If $A = \begin{pmatrix} 1 & 5 & 3 & 4 & 6 \\ 0 & 1 & 3 & 5 & 6 \end{pmatrix}$ and $b = \begin{pmatrix} 14 \\ 7 \end{pmatrix}$, then $x = (0, 2, 0, 1, 0)$ is a basic feasible solution with $B = \{2, 4\}$.

Ex If $A = \begin{pmatrix} 1 & 5 & 3 & 4 & 6 \\ 0 & 1 & 3 & 5 & 6 \end{pmatrix}$ and $b = \begin{pmatrix} 10 \\ 2 \end{pmatrix}$,
then $x = (0,2,0,0,0)$ is a basic feasible solution with four different choices for B:
$B = \{1,2\}, \{2,3\}, \{2,4\}$, or $\{2,5\}$.

Moral A basic feasible solution (bfs) x does not determine the basis B.

By contrast,

Prop 4.2.2 A basis B determines at most one bfs x.

Why? In the above example, if we set $B = \{1,4\}$, then the bfs x must satisfy

$$x = (x_1, 0, 0, x_4, 0).$$

$$\begin{pmatrix} 10 \\ 2 \end{pmatrix} = b = Ax = A_B \begin{pmatrix} x_1 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1 & 4 \\ 0 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_4 \end{pmatrix} = \begin{pmatrix} x_1 + 4x_4 \\ 5x_4 \end{pmatrix}$$

$\Longrightarrow \begin{pmatrix} x_1 \\ x_4 \end{pmatrix} = \begin{pmatrix} 42/5 \\ 2/5 \end{pmatrix}$, using the fact that A_B is invertible.

So $x = (42/5, 0, 0, 2/5, 0)$.

Ex If $A = \begin{pmatrix} 1 & 5 & 3 & 4 & 6 \\ 0 & 1 & 3 & 5 & 6 \end{pmatrix}$ and $b = \begin{pmatrix} 10 \\ 2 \end{pmatrix}$, then does $B = \{3, 4\}$ yield a bfs x?

Ans $x = (0, 0, x_3, x_4, 0)$

$$\begin{pmatrix} 10 \\ 2 \end{pmatrix} = b = Ax = A_B \begin{pmatrix} x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 3 & 4 \\ 3 & 5 \end{pmatrix} \begin{pmatrix} x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 3x_3 + 4x_4 \\ 3x_3 + 5x_4 \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 42/3 \\ -8 \end{pmatrix}.$$

No, $B = \{3, 4\}$ does not yield a bfs since the corresponding $x = (0, 0, 42/3, -8, 0)$ is not nonnegative.

Ex Does $B = \{3, 5\}$ yield a bfs x?

Ans No, B is not even a basis since $A_B = \begin{pmatrix} 3 & 6 \\ 3 & 6 \end{pmatrix}$ is singular.

Thm 4.2.3 If an optimal solution exists to
Maximize $c^T x$
subject to $Ax = b$ $\quad$ (A size $m \times n$)
$x \geq 0$,
then there is also a bfs that is optimal.

Proof 1 Follows from the proof of correctness of the simplex method.

Proof 2 Follow since each vertex of the feasible region corresponds to a bfs.

Impractical algorithm for solving linear programs

Consider all $\binom{n}{m}$ subsets $B \subseteq \{1, \ldots, n\}$ of size m,
see if B corresponds to a bfs x,
take the max over all $c^T x$.

Basic feasible solutions — Geometry

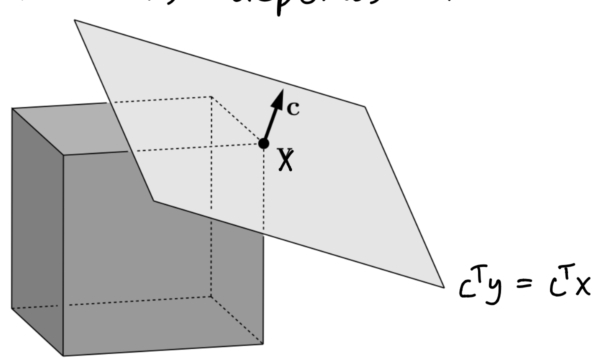
Recall Maximize $c^T x$
subject to $Ax = b$ (A size $m \times n$)
$x \geq 0$

Def A feasible solution $x \in \mathbb{R}^n$ is <u>basic</u> if there is an m-element set $B \subseteq \{1, 2, \ldots, n\}$ such that
- the square matrix A_B is nonsingular, and
- $x_j = 0$ for all $j \notin B$.

Prop 4.2.2 A basis B determines at most one bfs x.

Thm 4.2.3 If an optimal solution exists, then an optimal bfs exists.

Rmk Nothing about a bfs depends on c.



Def x is a <u>vertex</u> of a convex polyhedron $P \subseteq \mathbb{R}^n$ if there is some $c \in \mathbb{R}^n$ with $c^T x > c^T y$ for all $y \in P \setminus \{x\}$.

Thm 4.4.1 Given a linear program in equational form, x is a vertex of the feasible region $\iff$ x is a bfs.

Pf ($\Rightarrow$) Follows from Thm 4.2.3, with c being the vector showing x is a vertex.

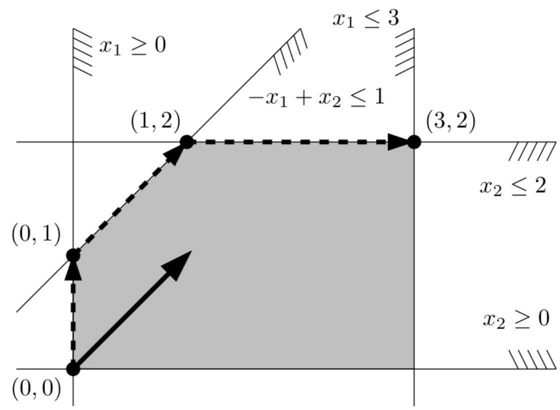
($\Leftarrow$) Let x be a bfs with basis B.
Define $c \in \mathbb{R}^n$ by $c_j = \begin{cases} 0 & j \in B \\ -1 & j \notin B. \end{cases}$

Note $c^T x = 0$, and Prop 4.2.2 implies $c^T y < 0$ for all feasible $y \neq x$.
Hence x is a vertex of the feasible region.

The simplex method — Introductory example

Maximize $x_1 + x_2$

subject to
$$\begin{aligned} -x_1 + x_2 &\le 1 \\ x_1 &\le 3 \\ x_2 &\le 2 \\ x_1, x_2 &\ge 0 \end{aligned}$$



Equational form

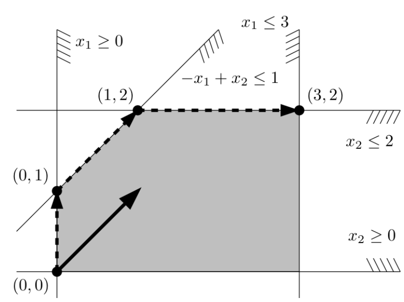
Maximize $x_1 + x_2$

subject to
$$\begin{aligned} -x_1 + x_2 + x_3 \qquad\qquad &= 1 \\ x_1 \qquad + x_4 \qquad &= 3 \\ x_2 \qquad\qquad + x_5 &= 2 \\ x_1, x_2, x_3, x_4, x_5 &\ge 0 \end{aligned}$$

$$A = \begin{pmatrix} -1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Simplex tableau A

$$\begin{aligned} x_3 &= 1 + x_1 - x_2 \\ x_4 &= 3 - x_1 \\ x_5 &= 2 \qquad - x_2 \\ \hline z &= \quad\; x_1 + x_2 \end{aligned}$$

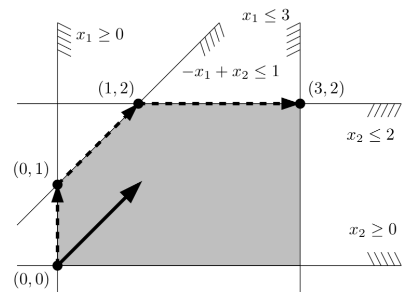


Plug in $x_1 = x_2 = 0$ to get a bfs with basis $B = \{3, 4, 5\}$ and value $z = 0$.

Increase z by (arbitrarily) deciding to increase x_2 while fixing $x_1 = 0$.

We are most limited by the equation $x_3 = 1 + x_1 - x_2$, which we can rewrite as $x_2 = 1 + x_1 - x_3$

Simplex tableau B

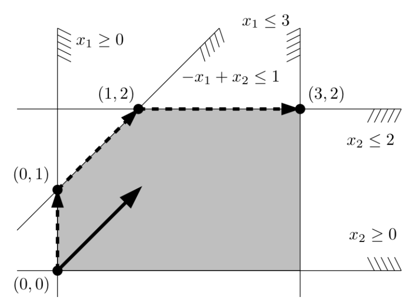


$$
\begin{aligned}
x_2 &= 1 + x_1 - x_3 \\
x_4 &= 3 - x_1 \\
x_5 &= 1 - x_1 + x_3 \\
\hline
z &= 1 + 2x_1 - x_3
\end{aligned}
$$

Plug in $x_1 = x_3 = 0$ to get a bfs with basis $B = \{2, 4, 5\}$ and value $z = 1$.

Increase z by increasing x_1.
Limited by $x_5 = 1 - x_1 + x_3$, i.e. $x_1 = 1 + x_3 - x_5$

Simplex tableau C



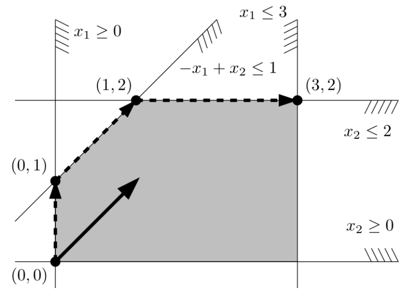
$$
\begin{aligned}
x_1 &= 1 + x_3 - x_5 \\
x_2 &= 2 \qquad\;\; - x_5 \\
x_4 &= 2 - x_3 + x_5 \\
\hline
z &= 3 + x_3 - 2x_5
\end{aligned}
$$

Plug in $x_3 = x_5 = 0$ to get a bfs with basis $B = \{1, 2, 4\}$ and value $z = 3$.

Increase z by increasing x_3.
Limited by $x_4 = 2 - x_3 + x_5$, i.e. $x_3 = 2 - x_4 + x_5$.

Simplex tableau D

$$
\begin{aligned}
x_1 &= 3 - x_4 \\
x_2 &= 2 \qquad\quad - x_5 \\
x_3 &= 2 - x_4 + x_5 \\
\hline
z &= 5 - x_4 - x_5
\end{aligned}
$$



Plug in $x_4 = x_5 = 0$ to get a bfs with basis $B = \{1,2,3\}$ and value $z = 5$.

This is optimal, and moreover gives a proof of optimality, since any feasible solution satisfies $z = 5 - x_4 - x_5$ with $x_4, x_5 \geq 0$.

<u>Rmk</u> Pictures should really be in $\mathbb{R}^5$, not $\mathbb{R}^2$.

<u>Rmk</u> Fewer steps if we had first pivoted over x_1.

Exception handling: Unboundedness

$$\begin{aligned} \text{Maximize} \quad & x_1 \\ \text{subject to:} \quad & x_1 - x_2 \le 1 \\ & -x_1 + x_2 \le 2 \\ & x_1, x_2 \ge 0 \end{aligned}$$

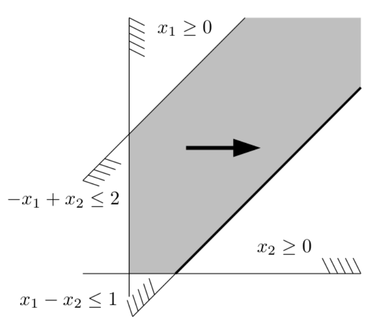
Equational form

$$\begin{aligned} \text{Maximize} \quad & x_1 \\ \text{subject to:} \quad & x_1 - x_2 + x_3 = 1 \\ & -x_1 + x_2 + x_4 = 2 \\ & x_1, x_2, x_3, x_4 \ge 0 \end{aligned}$$

Simplex tableau

$$\begin{array}{rcl} x_3 &=& 1 - x_1 + x_2 \\ x_4 &=& 2 + x_1 - x_2 \\ \hline z &=& x_1 \end{array} \xrightarrow[\text{pivot on } x_1]{} \begin{array}{rcl} x_1 &=& 1 + x_2 - x_3 \\ x_4 &=& 3 - x_3 \\ \hline z &=& 1 + x_2 - x_3 \end{array}$$

$\xrightarrow[\text{pivot on } x_2]{}$ Feasible ray $\{(1+x_2, x_2, 0, 3) : x_2 \ge 0\}$ with unbounded objective function $1 + x_2$.



$$\begin{aligned} \text{Maximize} \quad & x_1 \\ \text{Subject to:} \quad & x_1 - x_2 \le 1 \\ & -x_1 + x_2 \le 2 \\ & x_1, x_2 \ge 0 \end{aligned}$$

Exception handling: Degeneracy

Equational form

Maximize x_2

subject to: $-x_1 + x_2 \le 0$

$x_1 \le 2$

$x_1, x_2 \ge 0$

Maximize x_2

subject to $-x_1 + x_2 + x_3 = 0$

$x_1 + x_4 = 2$

$x_1, x_2, x_3, x_4 \ge 0$

Simplex tableau

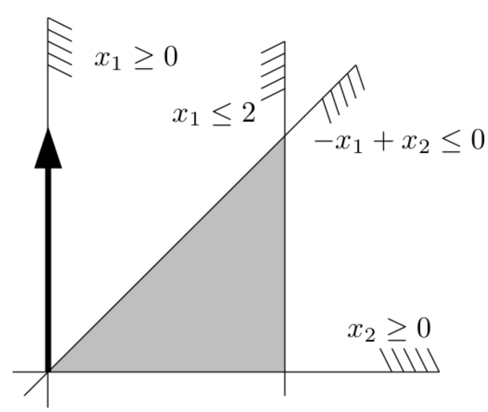
$$\begin{aligned} x_3 &= x_1 - x_2 \\ x_4 &= 2 - x_1 \\ \hline z &= x_2 \end{aligned} \xrightarrow{\text{Pivot } x_2} \begin{aligned} x_2 &= x_1 - x_3 \\ x_4 &= 2 - x_1 \\ \hline z &= x_1 - x_3 \end{aligned}$$

Feasible solution: $(0,0,0,2)$

Feasible solution: $(0,0,0,2)$

$$\xrightarrow{\text{Pivot } x_1} \begin{aligned} x_1 &= 2 - x_4 \\ x_2 &= 2 - x_3 - x_4 \\ \hline z &= 2 - x_3 - x_4 \end{aligned}$$

Optimal solution: $(2, 2, 0, 0)$



Remark There are ways to prevent cycling.

Exception handling: Infeasibility

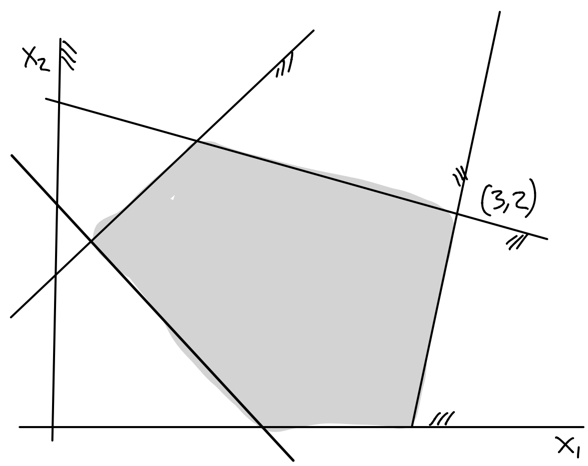
You get a feasible basis for free in

Maximize $c^T x$

subject to $Ax \leq b$, $x \geq 0$

by adding slack variables to get to equational form and then letting the basis be the slack variables.

But in general, recall finding a feasible solution is as hard as finding an optimal one!



Maximize $\quad x_1 + 2x_2$

subject to
$$x_1 + 3x_2 + x_3 = 4$$
$$2x_2 + x_3 = 2$$
$$x_1, x_2, x_3 \ge 0$$

Note $(x_1, x_2, x_3) = (0,0,0)$ is not feasible.

Auxilliary problem to find feasible solution via simplex method

Maximize $\quad -x_4 - x_5$

subject to
$$x_1 + 3x_2 + x_3 + x_4 = 4$$
$$2x_2 + x_3 + x_5 = 2$$
$$x_1, x_2, x_3, x_4, x_5 \ge 0$$

The objective value is $0 \iff$ there is a feasible solution to the original problem.

Simplex tableau

$$\begin{array}{l} x_4 = 4 - x_1 - 3x_2 - x_3 \\ x_5 = 2 - 2x_2 - x_3 \\ \hline z = -6 + x_1 + 5x_2 + 2x_3 \end{array} \xrightarrow[\text{Pivot } x_1]{} \begin{array}{l} x_1 = 4 - 3x_2 - x_3 - x_4 \\ x_5 = 2 - 2x_2 - x_3 \\ \hline z = -2 + 2x_2 + x_3 - x_4 \end{array}$$

$$\xrightarrow[\text{Pivot } x_3]{} \begin{array}{l} x_1 = 2 - x_2 - x_4 + x_5 \\ x_3 = 2 - 2x_2 - x_5 \\ \hline z = -x_4 - x_5 \end{array}$$

Anxilliary optimal solution $(2,0,2,0,0)$ yields the basic feasible solution $(2,0,2)$ of the original problem.

Original

Maximize $x_1 + 2x_2$

subject to $x_1 + 3x_2 + x_3 = 4$

$2x_2 + x_3 = 2$

$x_1, x_2, x_3 \geq 0$

Tableau

$$\begin{aligned} x_1 &= 2 - x_2 \\ x_3 &= 2 - 2x_2 \\ \hline z &= 2 + x_2 \end{aligned}$$

$\xrightarrow{}$ Pivot x_2

$$\begin{aligned} x_1 &= 1 + \tfrac{1}{2} x_3 \\ x_2 &= 1 - \tfrac{1}{2} x_3 \\ \hline z &= 3 - \tfrac{1}{2} x_3 \end{aligned}$$

Optimum solution $(1, 1, 0)$ with value 3.

Simplex tableus in general

Maximize $z = c^T x$ subject to $Ax = b$ and $x \geq 0$, A size $m \times n$.

Recall a feasible basis is a m-element set $B \subseteq \{1, 2, \ldots, n\}$ with A_B nonsingular and the (unique) solution $A_B x_B = b$ nonnegative.

Ex

$$
\begin{array}{ll}
\text{Maximize} & x_1 + x_2 \\
\text{subject to} & -x_1 + x_2 + x_3 \qquad\qquad = 1 \\
& x_1 \qquad\qquad + x_4 \quad = 3 \\
& \quad x_2 \qquad\qquad + x_5 = 2 \\
& x_1, x_2, x_3, x_4, x_5 \geq 0
\end{array}
\qquad
A = \begin{pmatrix} -1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}
$$

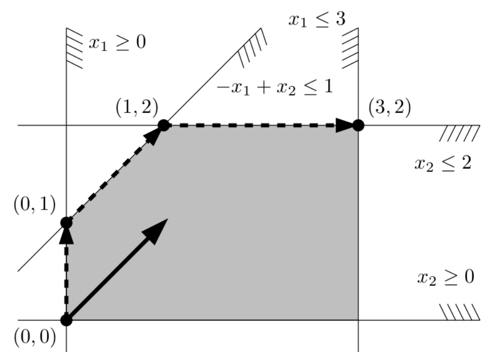
Def The simplex tableau determined by feasible basis B is

$$
\begin{aligned}
x_B &= p + Q x_N \\
z &= z_0 + r^T x_N
\end{aligned}
$$

where x_B is the vector of basic variables, $N = \{1, 2, \ldots, n\} \setminus B$, x_N is the vector of nonbasic variables, $p \in \mathbb{R}^m$, $r \in \mathbb{R}^{n-m}$, Q $m \times (n-m)$ matrix, $z_0 \in \mathbb{R}$.

Simplex tableau

$$\begin{aligned} x_1 &= 1 + x_3 - x_5 \\ x_2 &= 2 - x_5 \\ x_4 &= 2 - x_3 + x_5 \\ \hline z &= 3 + x_3 - 2x_5 \end{aligned}$$



Fake Lemma 5.5.1

$Q = -A_B^{-1} A_N$, $\quad p = A_B^{-1} b$,

$z_0 = c^T x_B A_B^{-1} b$, $\quad r = c_N - (c_B^T A_B^{-1} A_N)^T$

Real Lemma 5.5.1

Don't memorize these formulas, just know they exist and depend on A_B^{-1}.

Pf

$$\begin{pmatrix} x_B = p + Q x_N \\ z = z_0 + r^T x_N \end{pmatrix}$$

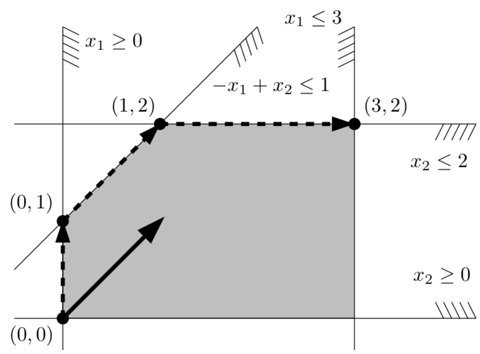
Rewrite $Ax = b$ as $A_B x_B + A_N x_N = b$, or $A_B x_B = b - A_N x_N$, giving $x_B = A_B^{-1}(b - A_N x_N)$.

$$\begin{aligned} z = c^T x &= c_B^T x_B + c_N^T x_N \\ &= c_B^T \left(A_B^{-1} (b - A_N x_N) \right) + c_N^T x_N \\ &= c_B^T A_B^{-1} b + \left(c_N^T - c_B^T A_B^{-1} A_N \right) x_N. \end{aligned}$$

$$\left(\begin{array}{l} x_B = p + Q x_N \\ z = z_0 + r^T x_N \end{array}\right)$$

Simplex tableau

$$\begin{aligned} x_1 &= 1 + x_3 - x_5 \\ x_2 &= 2 - x_5 \\ x_4 &= 2 - x_3 + x_5 \\ \hline z &= 3 + x_3 - 2x_5 \end{aligned}$$



Rmk If $r \leq 0$ then the corresponding bfs is optimal.

Rmk A nonbasic variable x_v may enter the basis $\iff$ its coefficient in r is positive.

Rmk When pivoting, the leaving variable x_u satisfies $q_{uv} < 0$ and
$$\frac{-p_u}{q_{uv}} = \min\left\{ \frac{-p_i}{q_{iv}} \;\middle|\; q_{iv} < 0 \right\}.$$

(Could be more than one choice of leaving variable.)

Simplex method — Computations

Rmk Computer implementations don't store the full tableau, only B, A_B^{-1} and $p = A_B^{-1} b$

Simplex tableau

$$x_1 = 1 + x_3 - x_5$$
$$x_2 = 2 \qquad - x_5$$
$$x_4 = 2 - x_3 + x_5$$
$$z = 3 + x_3 - 2x_5$$

$B = \{1, 2, 4\}$

$$\begin{pmatrix} 1 & -1 \\ 0 & -1 \\ -1 & 1 \end{pmatrix} = Q = A_B^{-1} A_N$$

$$(1, 2, 2)^T = p = A_B^{-1} b$$

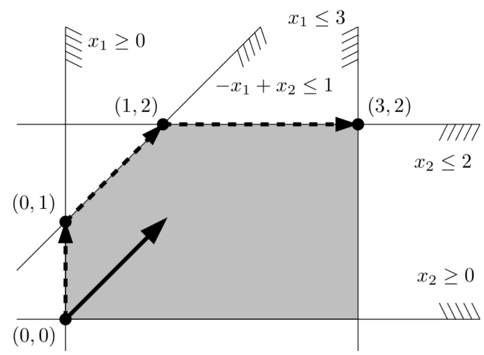
Rmk Due to numerical imprecision, the explicit inverse A_B^{-1} is not the best choice: often A_B^{-1} is represented by an LU-factorization of A_B.

Rmk In this revised simplex method, a pivot step takes time $O(m^2)$ instead of $O(mn)$ operations with the full tableau.

Pivot rules

Simplex tableau

$$\begin{aligned} x_3 &= 1 + x_1 - x_2 \\ x_4 &= 3 - x_1 \\ x_5 &= 2 - x_2 \\ \hline z &= x_1 + x_2 \end{aligned}$$



Dantzig's rule / Largest coefficient (in z row)
Maximizes improvement of z per unit increase in entering variable.

Largest increase (of z)
More expensive, but locally greedy.

Steepest edge Max $\dfrac{c^T(x_{new} - x_{old})}{\|x_{new} - x_{old}\|}$
Champion among pivot rules in practice!
Efficient approximate implementation: "Devex"

Bland's rule smallest index for both entering and leaving variables. Prevents cycling.

Random edge Random methods lead to best provable bounds for expected simplex method efficiency.

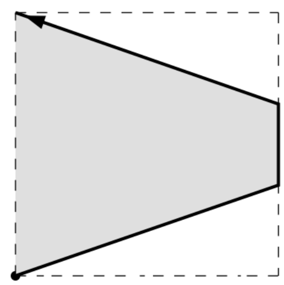
Efficiency of the simplex method

In the worst case, the simplex method runs in exponential time.

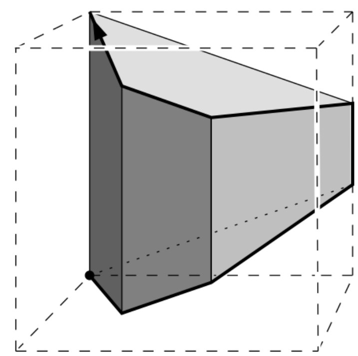
($n =$ # variables
$m =$ # equations)

Klee-Minty cubes

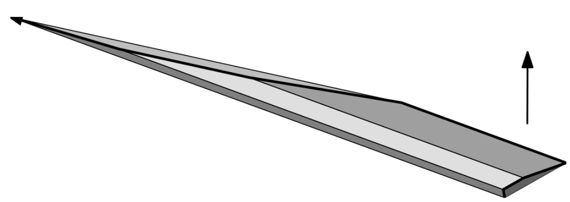
With some choice of pivots, the simplex method will require $2^n - 1$ pivot steps.



$n = 2$



$n = 3$



$$\begin{array}{ll} \text{Maximize} & 9x_1 + 3x_2 + x_3 \\ \text{subject to} & x_1 \leq 1 \\ & 6x_1 + x_2 \leq 9 \\ & 18x_1 + 6x_2 + x_3 \leq 81 \\ & x_1, x_2, x_3 \geq 0. \end{array}$$

With Dantzig's original pivot rule (largest coefficient), this Klee-Minty cube again requires $2^n - 1$ pivot steps.

Very slow examples of a similar type have been constructed for all non-random pivot rules above.

Goal which has never been attained Design a pivot rule and prove the # of pivot steps is bounded by a polynomial in n, m.

Positive result Randomly order all variables. Use Bland's rule for entering variable, lexicographic rule for leaving variable. Then

$$\mathbb{E}(\#\text{ pivot steps}) \le e^{C\sqrt{n \ln(n)}}.$$

expectation over all variable orderings

Rmk Better than 2^n, much worse than polynomial.

Clairvoyant pivot "rule" Pretend you can see the future — take the shortest path (fewest # of pivot steps).

Positive result Clairvoyant rule needs at most $n^{1+\ln(n)}$ steps.

Hirsch conjecture Clairvoyant rule needs only $O(n)$ steps — disproven by Francisco Santos, 2011.

Polynomial Hirsch conjecture — still open.

In practice, simplex method is impressively fast.

Positive result A "random" linear program in equational form requires only m^2 pivot steps with high probability.

Positive result An arbitrary linear program, after adding an insignificant amount of random noise to the coefficients, requires only polynomially many pivot steps with high probability.

Duality of linear programming

$$\begin{array}{ll} \text{Maximize} & 2x_1 + 3x_2 \\ \text{Subject to} & 4x_1 + 8x_2 \le 12 \\ & 2x_1 + x_2 \le 3 \\ & 3x_1 + 2x_2 \le 4 \\ & x_1, x_2 \ge 0 \end{array}$$

Without computing the optimum, we can infer:

$$2x_1 + 3x_2 \le 4x_1 + 8x_2 \le 12$$

by the nonnegativity constraints

Better: $2x_1 + 3x_2 \le \frac{1}{2}(4x_1 + 8x_2) \le \frac{1}{2}(12) = 6$

Better: $2x_1 + 3x_2 = \frac{1}{3}(4x_1 + 8x_2 + 2x_1 + x_2) \le \frac{1}{3}(12 + 3) = 5$

How good of an upper bound can we get in this way?

Derived from the constraints, we want an inequality

$$d_1 x_1 + d_2 x_2 \le h$$

with $d_1 \ge 2$, $d_2 \ge 3$ and h as small as possible.

$$
\begin{array}{ll}
\text{Maximize} & 2x_1 + 3x_2 \\
\text{subject to} & 4x_1 + 8x_2 \le 12 \\
& 2x_1 + x_2 \le 3 \\
& 3x_1 + 2x_2 \le 4 \\
& x_1, x_2 \ge 0
\end{array}
$$

Derived from the constraints, we want an inequality

$$d_1 x_1 + d_2 x_2 \le h$$

with $d_1 \ge 2$, $d_2 \ge 3$ and h as small as possible.

How do we get this? Consider

$$y_1(4x_1 + 8x_2) + y_2(2x_1 + x_2) + y_3(3x_1 + 2x_2) \le 12y_1 + 3y_2 + 4y_3$$

$$\| $$

$$(4y_1 + 2y_2 + 3y_3)x_1 + (8y_1 + y_2 + 2y_3)x_2 \quad \text{with } y_1, y_2, y_3 \ge 0.$$

Thus $d_1 = 4y_1 + 2y_2 + 3y_3$, $d_2 = 8y_1 + y_2 + 2y_3$, $h = 12y_1 + 3y_2 + 4y_3$.

To find the best y_1, y_2, y_3, we solve a dual linear program:

$$
\begin{array}{ll}
\text{Minimize} & 12y_1 + 3y_2 + 4y_3 \\
\text{subject to} & 4y_1 + 2y_2 + 3y_3 \ge 2 \\
& 8y_1 + y_2 + 2y_3 \ge 3 \\
& y_1, y_2, y_3 \ge 0
\end{array}
$$

How well does a dual linear program bound the original? Perfectly!

Dual LP has optimum $(y_1, y_2, y_3) = (\frac{5}{16}, 0, \frac{1}{4})$ with value 4.75

Primal LP has optimum $(x_1, x_2) = (\frac{1}{2}, \frac{5}{4})$ with value 4.75.

Primal LP

Maximize $2x_1 + 3x_2$

subject to

$$
\begin{aligned}
4x_1 + 8x_2 &\le 12 \\
2x_1 + x_2 &\le 3 \\
3x_1 + 2x_2 &\le 4. \\
x_1, x_2 &\ge 0
\end{aligned}
$$

Dual LP

Minimize $12y_1 + 3y_2 + 4y_3$

subject to

$$
\begin{aligned}
4y_1 + 2y_2 + 3y_3 &\ge 2 \\
8y_1 + y_2 + 2y_3 &\ge 3 \\
y_1, y_2, y_3 &\ge 0
\end{aligned}
$$

More generally, the dual of

maximize $c^T x$ subject to $Ax \le b$ and $x \ge 0$

is

minimize $b^T y$ subject to $A^T y \ge c$ and $y \ge 0$.

Weak duality theorem For feasible solutions x and y, we have $c^T x \le b^T y$.

If the primal is unbounded, then the dual is infeasible.

If the dual is unbounded, then the primal is infeasible.

from below

Proof $$\begin{aligned} c^T x &\le (A^T y)^T x \\ &= y^T A x \\ &\le y^T b \\ &= b^T y \end{aligned}$$

$$[y_1, \dots, y_m] \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} = b_1 y_1 + b_2 y_2 + \dots + b_m y_m = [b_1, \dots, b_m] \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

Strong duality theorem

Optimal feasible solutions satisfy $c^T x = b^T y$.

Feasibility vs optimality

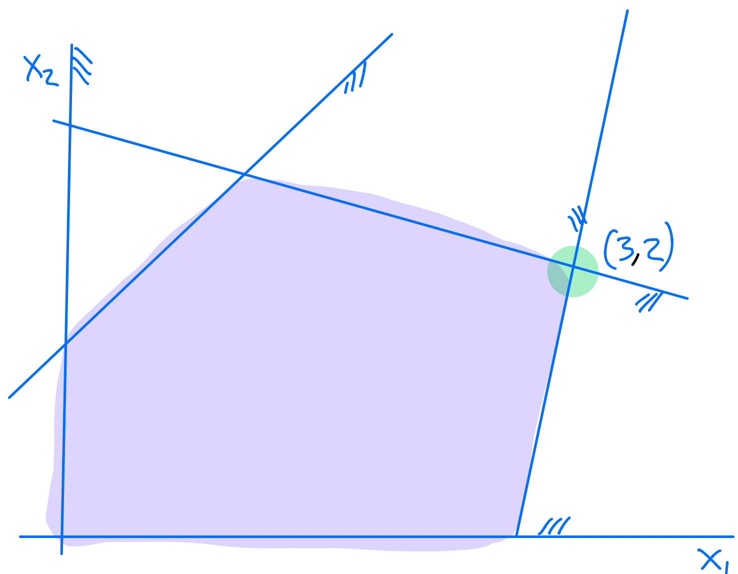
In some sense finding an optimal solution is no harder than finding a feasible solution.

First explanation: binary search

Ex Maximize $x_1 + x_2$

subject to

$$-x_1 + x_2 \le 1$$
$$x_1 + 6x_2 \le 15$$
$$4x_1 - x_2 \le 10$$
$$x_1, x_2 \ge 0$$



Second Explanation: Simplex method Phase I vs Phase II

Maximize $x_1 + 2x_2$

subject to
$$x_1 + 3x_2 + x_3 = 4$$
$$2x_2 + x_3 = 2$$
$$x_1, x_2, x_3 \geq 0$$

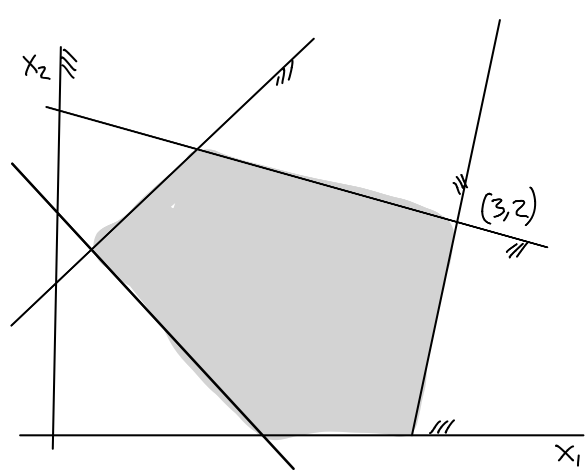
Note $(x_1, x_2, x_3) = (0,0,0)$ is not feasible.

Auxilliary problem to find feasible solution via simplex method

Maximize $-x_4 - x_5$

subject to
$$x_1 + 3x_2 + x_3 + x_4 = 4$$
$$2x_2 + x_3 + x_5 = 2$$

The objective value is $0 \iff$ there is a feasible solution to the original problem.



Third explanation, using duality

Primal: maximize $c^T x$ subject to $Ax \le b$ and $x \ge 0$.
Dual: minimize $b^T y$ subject to $A^T y \ge c$ and $y \ge 0$.

Weak duality: $c^T x \le b^T y$

Finding an optimal solution to

$$\text{Maximize } c^T x \quad \text{subject to } Ax \le b,\ x \ge 0$$

is the same as finding a <u>feasible</u> solution to

$$\begin{array}{ll} \cancel{\text{Maximize} \quad c^T x} & \\ \text{subject to} & Ax \le b \\ & A^T y \ge c \\ & c^T x \ge b^T y \\ & x \ge 0,\ y \ge 0 \end{array}$$

We know $c^T x \le b^T y$ for any feasible solutions to the primal and the dual, so adding $c^T x \ge b^T y$ as a constraint implies optimality.

Dualization recipe

Primal: Maximize $c^T x$ subject to $Ax \leq b$ and $x \geq 0$.
Dual: Minimize $b^T y$ subject to $A^T y \geq c$ and $y \geq 0$.

A size $m \times n$
Primal has n variables, m constraints
Dual has m variables, n constraints

Dualization Recipe

	Primal linear program	Dual linear program
Variables	$x_1, x_2, \ldots, x_n$	$y_1, y_2, \ldots, y_m$
Matrix	A	A^T
Right-hand side	$\mathbf{b}$	$\mathbf{c}$
Objective function	$\max \mathbf{c}^T \mathbf{x}$	$\min \mathbf{b}^T \mathbf{y}$
Constraints	ith constraint has $\leq$	$y_i \geq 0$
	$\geq$	$y_i \leq 0$
	$=$	$y_i \in \mathbb{R}$
	$x_j \geq 0$	jth constraint has $\geq$
	$x_j \leq 0$	$\leq$
	$x_j \in \mathbb{R}$	$=$

Primal: Maximize $c^T x$ subject to $Ax \le b$

(no nonnegativity constraints).

Dual: Minimize $b^T y$ subject to $A^T y = c$ and $y \ge 0$.

Primal: Max $3x_1 + 2x_2 + 4x_3$

subject to
$$
\begin{aligned}
2x_1 \qquad\quad - x_3 &\ge 4 \\
x_1 + x_2 + 3x_3 &= 7 \\
x_1 \le 0, \quad x_3 &\ge 0
\end{aligned}
$$

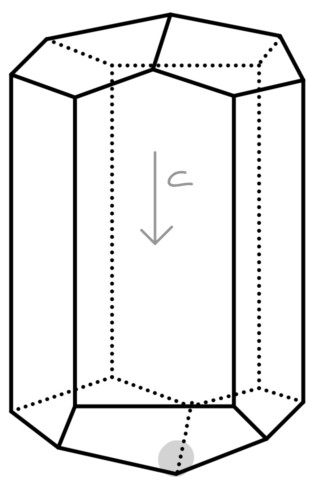
Dual: Min $4y_1 + 7y_2$

Subject to
$$
\begin{aligned}
2y_1 + y_2 &\le 3 \\
y_2 &= 2 \\
-y_1 + 3x_2 &\ge 4 \\
y_1 &\ge 0
\end{aligned}
$$

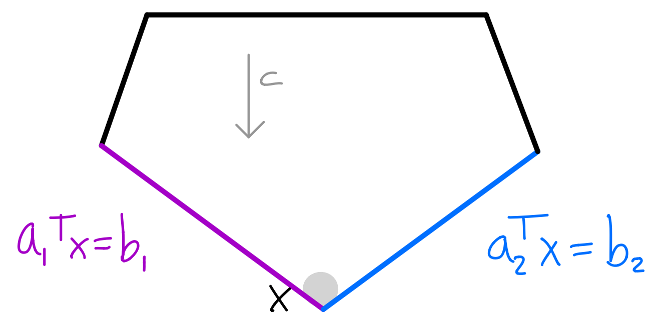
A physical interpretation of strong duality

Primal: Maximize $c^T x$ subject to $Ax \le b$ (no nonnegativity constraints).

Dual: Minimize $b^T y$ subject to $A^T y = c$ and $y \ge 0$.



Pretend c points downwards.



$$A = \begin{bmatrix} \cdots & a_1^T & \cdots \\ \cdots & a_2^T & \cdots \\ & \vdots & \\ \cdots & a_m^T & \cdots \end{bmatrix}$$

Forces: $c = y_1 a_1 + y_2 a_2$

We get $c = \sum_{i=1}^{m} y_i a_i = A^T y$.

(all other y_i's are 0)

So y is a feasible solution of the dual.

In $y^T(Ax-b)$, note

- $y_i = 0$ if the i^{th} face is not supporting.
- The i^{th} component of $Ax-b$ is zero if the i^{th} face is supporting.

Hence $y^T(Ax-b) = 0$, so

$$b^T y = y^T b = y^T A x = c^T x$$

This is strong duality.

Farkas lemma

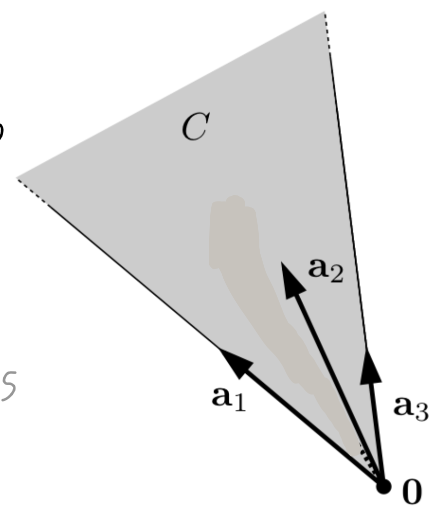
Let A be an $m \times n$ matrix, and let $b \in \mathbb{R}^m$. Then exactly one of the following two possibilities occurs.

(1) There exists $x \in \mathbb{R}^n$ with $Ax = b$ and $x \geq 0$, or

(2) There exists $y \in \mathbb{R}^m$ with $y^T A \geq 0^T$ and $y^T b < 0$.

<u>Def^n</u> The <u>convex cone</u> generated by $a_1, \ldots, a_n \in \mathbb{R}^m$ is

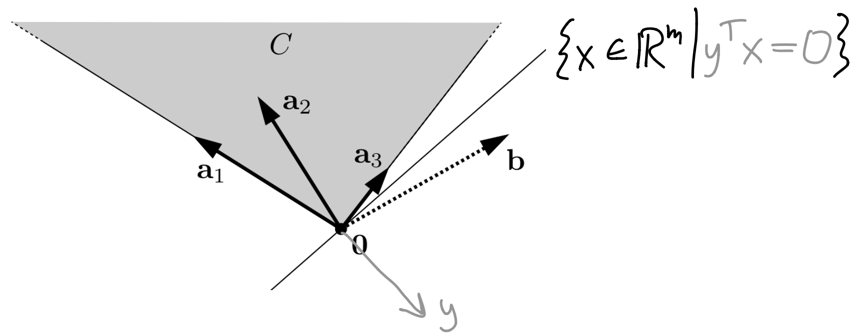
$$\{x_1 a_1 + \cdots + x_n a_n \mid x_1, \ldots, x_n \geq 0\}$$



Note $\{Ax \mid x \geq 0\}$ is the convex cone generated by the columns of A.

Option (1) of the Farkas lemma is when b is in the cone generated by the columns of A.

Option (2) of the Farkas lemma is when b is not in this cone.



This is an example <u>separation lemma</u> for convex sets. See also the Hahn-Banach theorem for infinite-dimensional Banach spaces in functional analysis.

See the section on a logical view of the Farkas lemma — our book makes an analogy with Hilbert's Nullstellensatz.

Our book gives proofs of the Farkas lemma using analysis, minimally infeasible systems, and Fourier-Motzkin elimination.

A variant of the Farkas lemma

Farkas: When does a system of linear equalities have a nonnegative solution?

Variant: When does a system of linear inequalities have a nonnegative solution?

Farkas lemma

Let A be an $m \times n$ matrix, and let $b \in \mathbb{R}^m$. Then exactly one of the following two possibilities occurs.

(1) There exists $x \in \mathbb{R}^n$ with $Ax = b$ and $x \geq 0$, or

(2) There exists $y \in \mathbb{R}^m$ with $y^T A \geq 0^T$ and $y^T b < 0$.

Variant of the Farkas lemma

The system $Ax \leq b$ has a nonnegative solution $x \geq 0$ if and only if every nonnegative $y \in \mathbb{R}^m$ with $y^T A \geq 0^T$ also satisfies $y^T b \geq 0$.

Proof Form the matrix $\overline{A} = (A \mid I_m)$

Note $Ax \leq b$ has a nonnegative solution if and only if $\overline{A}\overline{x} = b$ has a nonnegative solution.

(Analogy with slack variables)

By the Farkas Lemma, this is if and only if every $y \in \mathbb{R}^m$ with $y^T \overline{A} \geq 0^T$ satisfies $y^T b \geq 0$.

Means exactly $y^T A \geq 0^T$ and $y \geq 0$. $\square$

Proof of strong duality from the Farkas lemma

Variant of the Farkas lemma

The system $Ax \le b$ has a nonnegative solution $x \ge 0$ if and only if every nonnegative $y \in \mathbb{R}^m$ with $y^T A \ge 0^T$ also satisfies $y^T b \ge 0$.

Primal: Max $c^T x$ subject to $Ax \le b$ and $x \ge 0$

Dual: Min $b^T y$ subject to $A^T y \ge c$ and $y \ge 0$

Strong duality:

Optimal solutions x^*, y^* satisfy $c^T x^* = b^T y^*$.

Proof

(1) $Ax \le b, \quad c^T x \ge c^T x^*$ has a solution $x \ge 0$.

(2) $Ax \le b, \quad c^T x \ge c^T x^* + \varepsilon$ has no solutions $x \ge 0$ ~~for~~ any $\varepsilon > 0$.

Let $\hat{A} = \begin{pmatrix} A \\ -c^T \end{pmatrix} \in \mathbb{R}^{(m+1)\times n}$ and $\hat{b}_\varepsilon = \begin{pmatrix} b \\ -c^T x^* - \varepsilon \end{pmatrix} \in \mathbb{R}^{m+1}$.

(1) is $\hat{A} x \le \hat{b}_0$ and (2) is $\hat{A} x \le \hat{b}_\varepsilon$

Since (2) has no nonnegative solution, the variant of the Farkas lemma says there is some $\hat{y} = \begin{pmatrix} u \\ z \end{pmatrix} \in \mathbb{R}^{m+1}$ with $\hat{y}^T \hat{A} \geq 0$ but $\hat{y}^T \hat{b}_\varepsilon < 0$.

IE, $A^T u \geq zc$ and $b^T u < z(c^T x^* + \varepsilon)$.

Since (1) has a nonnegative solution, this very same $\hat{y}$ satisfies $\hat{y}^T \hat{b}_0 \geq 0$

IE $b^T u \geq z(c^T x^*)$.

Hence $z > 0$.

So $A^T(\frac{1}{z} u) \geq c$ and $b^T(\frac{1}{z} u) < c^T x^* + \varepsilon$

So $\frac{1}{z} u$ is a feasible solution of the dual with value less than $c^T x^* + \varepsilon$.

Letting $\varepsilon \to 0$ gives strong duality.

Other algorithms besides the simplex method

Ellipsoid method

- Polynomial time! Theoretically important.
  (Simplex method is not known to be,
  most pivot rules are known <u>not</u> to be polynomial)

- Not competitive in practice.
  Polynomial of high degree.

	Most problems	Worst case
Simplex method	Extremely fast	Exponentially bad
Ellipsoid method	High degree polynomial	High degree polynomial

- History: Invented 1970 by Shor, Judin, Nemirovski
  for nonlinear problems
  1979: Leonid Khachyian showed the ellipsoid method
  solves linear programs in polynomial time.
  Times interview of Georg Dantzig.

- Practical significance of this theoretical
  advancement: led to ...

Interior point methods

- Some of these are polynomial time
- Successfully competes with the simplex algorithm

Dual simplex method

- Perform simplex method on the dual.
- Implementation details can lead to crucial speed-ups.
- Particularly suitable when $n-m \ll m$.

Primal-dual method

- Through feasible solutions of dual, but not via pivots, instead via an auxilliary problem.
- Great for approximations of combinatorial optimizations.

Polynomial and strongly polynomial algorithms

Polynomial running time in terms of the input size — but what is the input size?

Notation

The bit size of an integer i is

$$\langle i \rangle = \lceil \log_2(|i|+1) \rceil + 1$$

This is the # of bits of i in binary, plus a sign.

Ex $\langle 1234 \rangle = 11 + 1 = 12$

since 1234 is 10011010010 in binary

For $r = p/q$ rational, we define $\langle r \rangle = \langle p \rangle + \langle q \rangle$.

Similarly for vectors and matrices.

A linear program Max $c^T x$ subject to $Ax \le b$ with rational entries has size $\langle A \rangle + \langle b \rangle + \langle c \rangle$.

Ex $A = \begin{bmatrix} 3 & -1 & 2 \\ 5 & 7 & 4 \end{bmatrix}$ $b = \begin{bmatrix} 2 \\ 5 \end{bmatrix}$ $c = \begin{bmatrix} 1{,}000{,}521 \\ 3{,}012{,}554 \\ 19{,}728{,}213 \end{bmatrix}$

Def An algorithm for linear programming is ***polynomial*** if there is a polynomial $p(x)$ such that for all linear programs with rational coefficients, the algorithm requires at most $p(\langle A\rangle + \langle b\rangle + \langle c\rangle)$ steps.

Steps are in terms of bit operations — adding two k bit integers requires at least k steps!

Digression: Gaussian Elimination

Solve $Ax=b$, A size $n\times n$, rational entries.

At most Cn^3 arithmetic operations.

Naive implementations can produce entries of size 2^n times larger, giving an exponential algorithm!

But smarter implementations are indeed polynomial.

Ellipsoid and interior point methods are polynomial, simplex method is not.

Strongly polynomial algorithms

Strongly polynomial algorithms are polynomial in the sense defined above, and <u>also</u> are polynomial in the # of arithmetic operations.
(More accurate representation of running time)

Gaussian elimination is strongly polynomial:
arithmetic operations is bounded by Cn^3, same for inputs with 10 bits or 1,000,000 bits.

A strongly polynomial algorithm for linear programming would be polynomial (as defined above) and require at most $p(n+m)$ arithmetic operations for some polynomial p.

No strongly polynomial algorithm for linear programming is known!

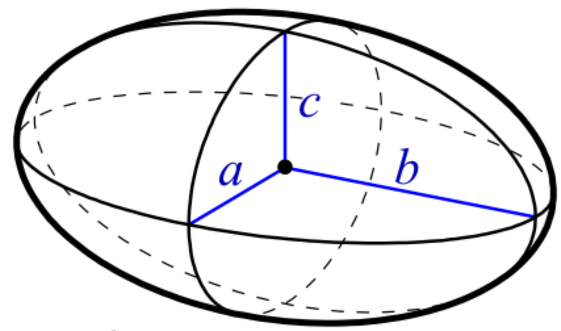
Ellipsoid method is not: For all $K \in \mathbb{N}$, we can find LP with 2 variables, 2 constraints, and ellipsoid method requires $\geq K$ arithmetic steps!

The bit sizes in these examples go to ∞ with K:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad \langle A \rangle \to \infty$$

$$\text{as } K \to \infty$$

The Ellipsoid Method



From wikipedia

- First polynomial time algorithm for linear programming
- Slow in practice
- We'll show how to find a feasible solution to $Ax \leq b$.
  Recall:

In some sense finding an optimal solution is no harder than finding a feasible solution.

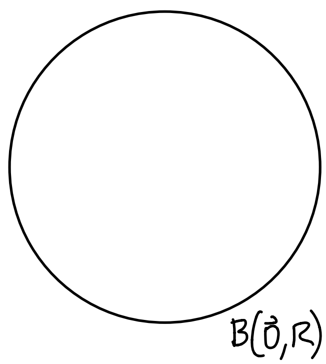
Finding an optimal solution to
$$\text{Maximize } c^T x \text{ subject to } Ax \leq b,\ x \geq 0$$
is the same as finding a feasible solution to

$$\cancel{\text{Maximize } c^T x}$$
$$\begin{aligned} \text{subject to} \quad & Ax \leq b \\ & A^T y \geq c \\ & c^T x \geq b^T y \\ & x \geq 0,\ y \geq 0 \end{aligned}$$

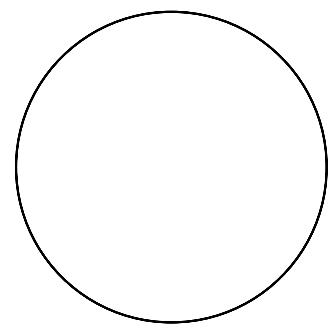
Analogy

Find feasible solution to $Ax \le b$

Find lion in Sahara



Find ε-ball lion in Sahara



Let $\varphi = \langle A \rangle + \langle b \rangle$ be the input size of $Ax \le b$.

Let $\eta = 2^{-5\varphi}$, $\varepsilon = 2^{-6\varphi}$.

Then $Ax \le b$ has a solution

$$\Longleftrightarrow$$

$Ax \le b + \eta$ has a solution (indeed a full ε-ball).

Aside: Also, $Ax \le b$ has a solution

$$\Longleftrightarrow$$

$Ax \le b,\ -2^{\varphi} \le x_1 \le 2^{\varphi},\ \ldots,\ -2^{\varphi} \le x_n \le 2^{\varphi}$ does.

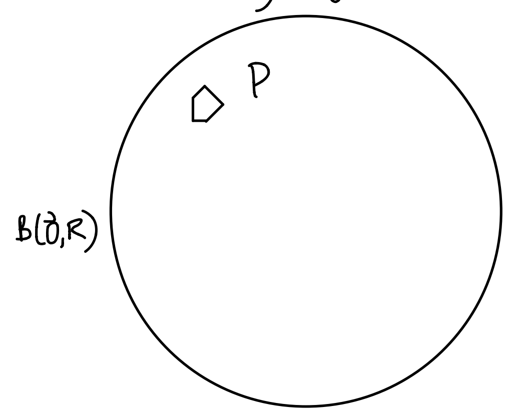
All solutions to the latter live in $B(\vec{0}, \underbrace{2^{\varphi}\sqrt{n}}_{=R})$.

Ellipsoid Algorithm

Inputs Matrix A, vector b, rational #'s $R > \varepsilon > 0$ with feasible region $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ contained in $B(\vec{0}, R)$.

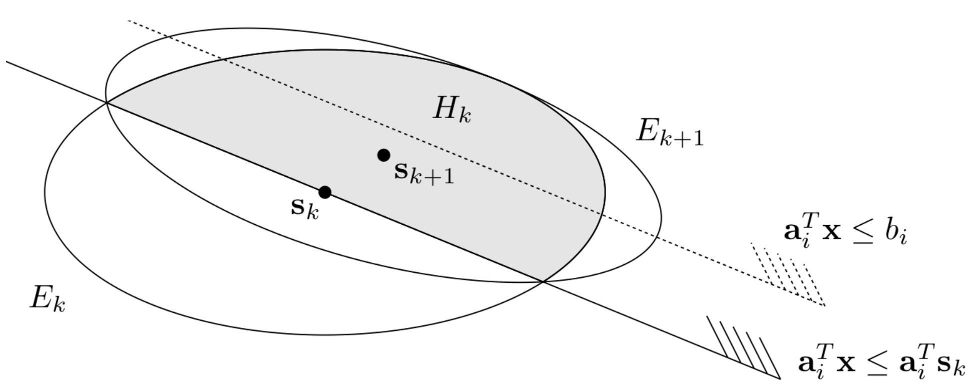
Output If P contains an ε-ball, return any $y \in P$. If not, return "no solution" or any $y \in P$.

We'll generate ellipsoids $B(\vec{0}, R) = E_0, E_1, E_2, E_3, \ldots$ with $P \subseteq E_k$ for all k.



Iterative step If the current ellipsoid E_k has center s_k in P, return that center and stop!

Else, find E_{k+1} from E_k as follows:



Stop if the volume of E_{k+1} is ever smaller than that of an ε-ball.

<u>Algorithm analysis</u> Why is it polynomial?

For all k, $\dfrac{\text{volume}(E_{k+1})}{\text{volume}(E_k)} \le e^{-1/(2n+2)}$.

Hence after $\lceil n(2n+2)\ln(R/\varepsilon) \rceil$ iterations, the volume of E_k is smaller than that of an ε-ball.

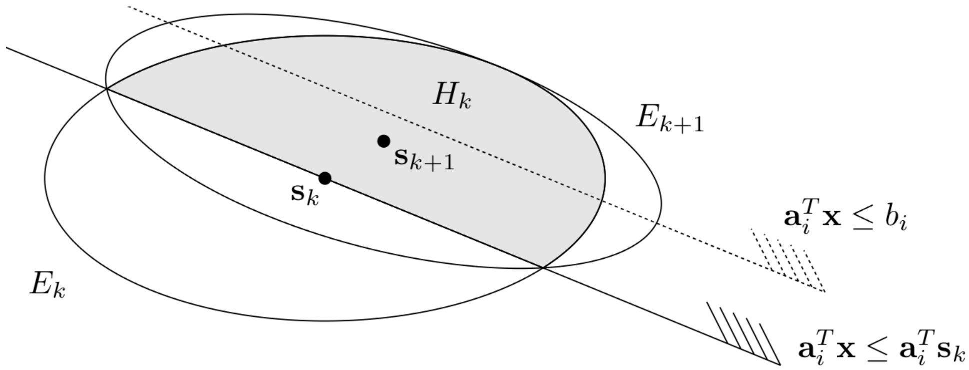
<u>Subtleties</u>

- Don't compute E_{k+1} exactly (square roots); use rational parameters and expand E_{k+1} slightly.

- If same inequality is violated too many times in a row, then the ellipsoids become too long.

- Ellipsoids could be replaced with other "rich-enough" families of convex sets, like simplices.

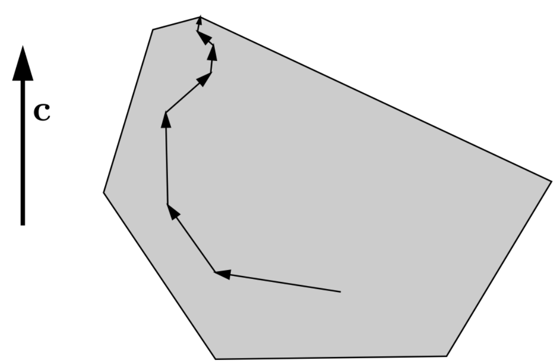
- All we need is a "separation oracle". We don't need to know $Ax \le b$. Could even have infinitely many constraints.



Semidefinite programming: Variable X is a matrix constrained to be positive semidefinite.
That is, $a^T X a \geq 0$ for <u>all</u> vectors a.

Interior point methods

- Used for nonlinear problems since 1950's
- Tested, without success, on linear problems in 1970's
- Press headlines in 1984: Narendra Karmakar, IBM, proof of polynomial time on linear problems
- Now competitive with the simplex method, especially on large problems (too large for the 1970's): rely on powerful routines for sparse systems of equations.



- Interior methods walk along the interior, until they hop to an exact optimum via a last rounding step.

(By contrast, the simplex method walks along the boundary, and the ellipse method encircles the set of feasible solutions from the outside.)

We avoid combinatorial intracacies of the boundary.

Types of interior point methods

Central path, potential reduction, affine scaling.

For each: primal, dual, primal-dual, or self-dual.

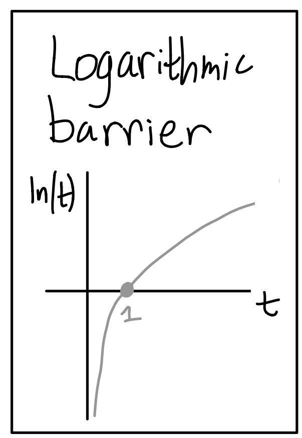
The Central Path

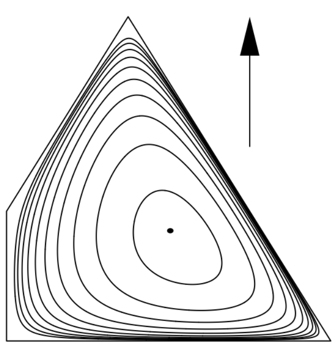
Max $c^T x$ subject to $Ax \le b$

For $\mu > 0$, define

$$f_\mu(x) = c^T x + \mu \cdot \sum_{i=1}^{m} \ln(b_i - a_i x)$$

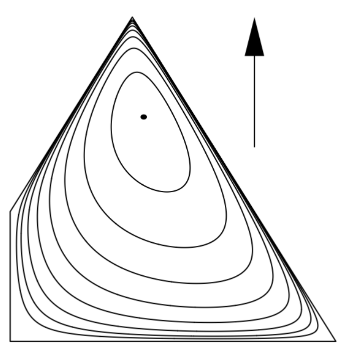
where a_i is the i-th row of A.



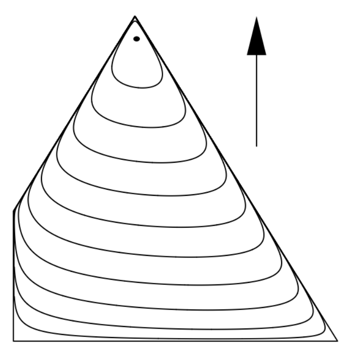


$\mu = 100$

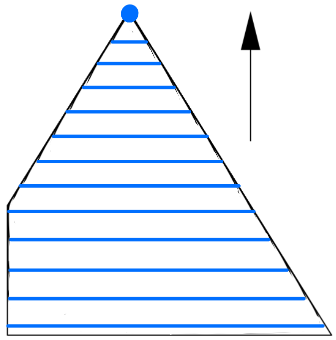
"farthest point from boundary"



$\mu = 1$



$\mu = \frac{1}{100}$



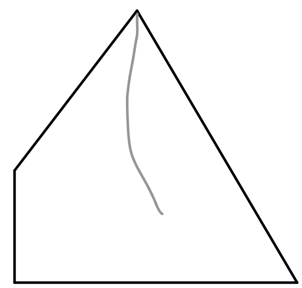
$\mu = 0$ (the limit)

Let $x^*(\mu)$ be the unique solution to

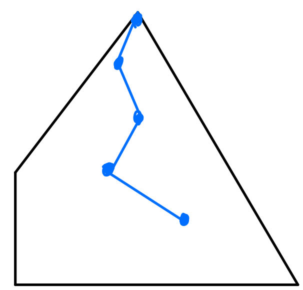
Max $f_\mu(x)$ subject to $Ax \le b$.

redundant

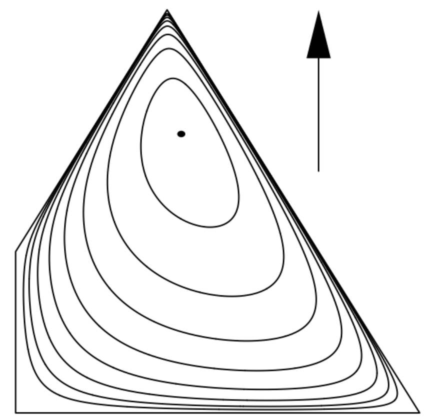
<u>Def$^{\underline{n}}$</u> The <u>central path</u> is the curve $x^*(\mu)$ for $\mu > 0$.



Actual algorithms will only approximately follow this path.

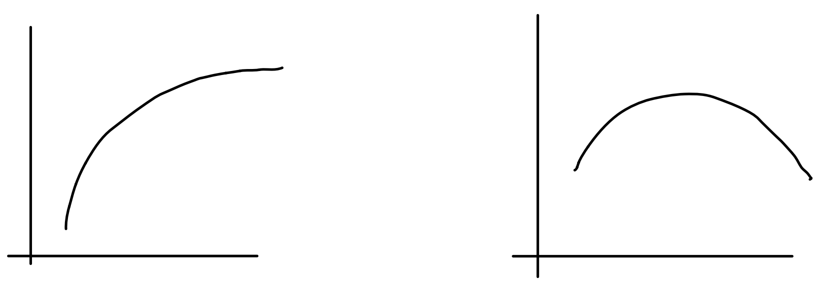


For $\mu > 0$ fixed, why is there a unique optimum $x^*(\mu)$?



- An optimum exists since we have a continuous function $f_\mu(x) = c^T x + \mu \cdot \sum_i \ln(b_i - a_i x)$ on a compact set $\{x : Ax \leq b \text{ and } f_\mu(x) \geq f_\mu(y)\}$.

- The optimum is unique since f_μ is strictly concave for $\mu > 0$.



The primal-dual central path (equational form)

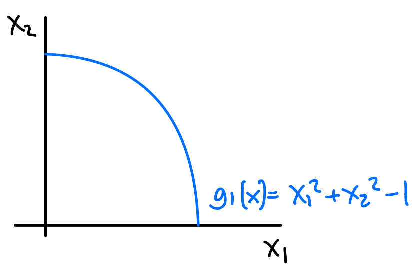
Maximize $c^T x$ subject to $Ax = b$, $x \geq 0$

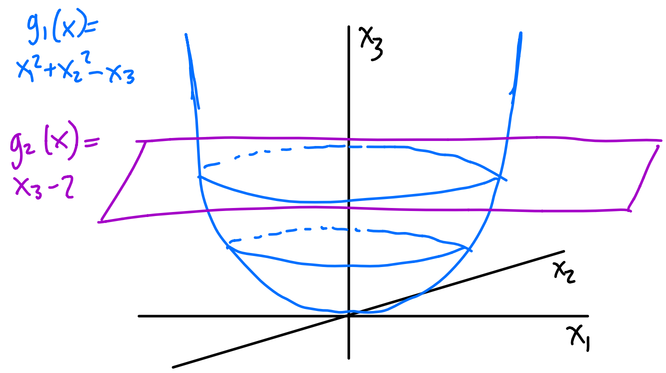
size $m \times n$

$$f_\mu(x) = c^T x + \mu \cdot \sum_{j=1}^{n} \ln x_j$$

__Lagrange Multipliers__ A maximum of $f(x)$ subject to $g_1(x) = g_2(x) = \dots = g_m(x) = 0$ satisfies $\nabla f(x) = \sum_{i=1}^{m} y_i \nabla g_i(x)$

(Functions are $\mathbb{R}^n \to \mathbb{R}$, gradients are row vectors.)





Apply Lagrange multipliers to

Maximize $f_\mu(x) = c^T x + \mu \cdot \sum_{j=1}^{m} \ln x_j$ subject to $Ax = b,\ x \geq 0$

$g_i(x) = a_i x - b_i$

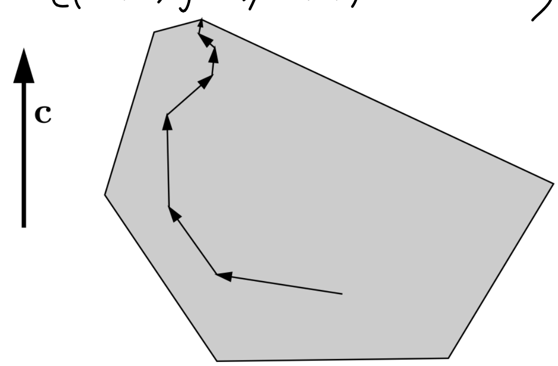
to get

$$c + \mu\left(\frac{1}{x_1}, \dots \frac{1}{x_n}\right) = \nabla f(x) = \sum_{i=1}^{m} y_i \nabla g_i(x) = \sum_{i=1}^{m} y_i a_i = A^T y$$

Introduce the nonnegative vector $s = \mu\left(\frac{1}{x_1}, \dots, \frac{1}{x_n}\right)$ to get

$$\boxed{\begin{aligned} Ax &= b \\ A^T y - s &= c \\ (s_1 x_1, s_2 x_2, \dots, s_n x_n) &= (\mu, \mu, \dots, \mu) \qquad \text{(Not linear)} \\ x, s &\geq 0 \qquad y \in \mathbb{R}^m \end{aligned}}$$

The primal-dual central path is $\left\{ \left(x^*(\mu), y^*(\mu), s^*(\mu)\right) \in \mathbb{R}^{2n+m} : \mu > 0 \right\}$



In some sense, Lagrange multipliers recover the duality of linear programming!

We started with

$$\text{Maximize } c^T x \quad \text{subject to} \quad Ax=b, \quad x \geq 0$$

whose dual is

$$\text{Minimize } b^T y \quad \text{subject to} \quad A^T y \geq c, \quad y \in \mathbb{R}^m.$$

We derived

$$\boxed{\begin{array}{rcll} Ax &=& b & \\ A^T y - s &=& c & \\ (s_1 x_1, s_2 x_2, \ldots, s_n x_n) &=& (\mu, \mu, \ldots, \mu) & \text{(Not linear)} \\ x, s &\geq& 0 & y \in \mathbb{R}^m \end{array}}$$

for $\mu > 0$, but setting $\mu = 0$ gives $(s_1 x_1, \ldots, s_n x_n) = (0, \ldots, 0)$, i.e.,

$$s^T x = 0 \quad \text{by nonnegativity.}$$

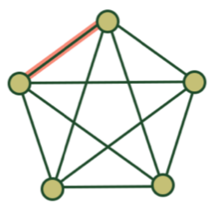
So
$$\begin{aligned} 0 &= s^T x \\ &= (A^T y - c)^T x \\ &= y^T A x - c^T x \\ &= y^T b - c^T x \qquad \text{since } Ax=b \end{aligned}$$

So x is a feasible solution of the primal, y is a feasible solution of the dual (with slack variables s_j), and $y^T b = c^T x$ implies these feasible solutions are optimal, by strong duality.

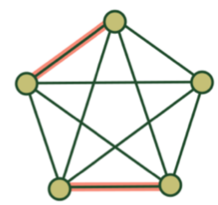
Matchings and Hall's Theorem

A *matching* in a graph $G=(V,E)$ is a set $E' \subseteq E$ of edges such that each vertex is incident to at most one edge in E'.

Matching:



Maximum matching:

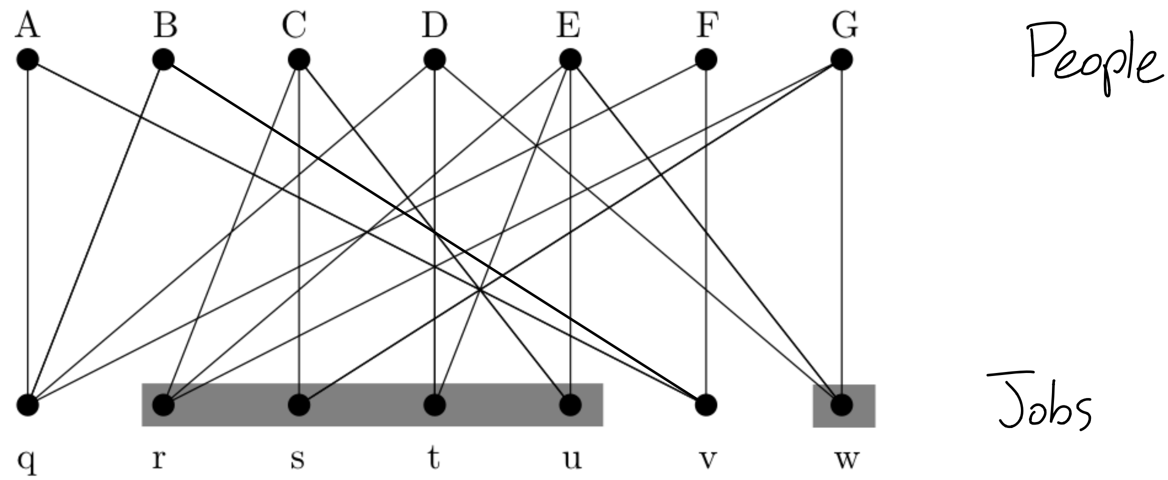


(Images thanks to:
Kelly Emmrich, Maria Gillespie, Shannon Golden, Rachel Pries)

Def A matching is *maximum* if it has the largest # of edges among all matchings in G.

Rmk Not every matching can be extended to a maximum one.

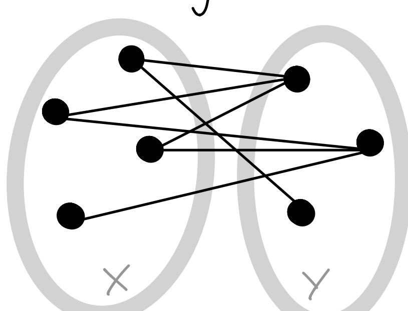
Example Is there a matching of people to jobs they would enjoy?

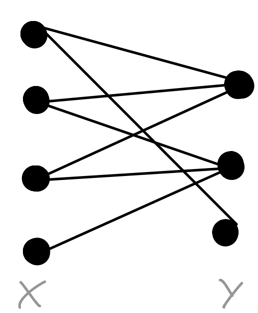


Neighborhood$(\{r, s, t, u, w\}) = \{C, D, E, G\}$.

Answer: No. No assignment can fill these 5 jobs with only 4 possible people.

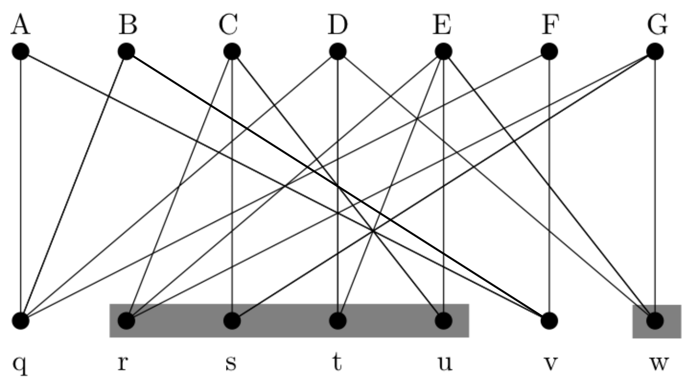
Def A graph $G=(V,E)$ is <u>bipartite</u> if its vertices can be divided into two disjoint groups $V = X \cup Y$ with all edges connecting between X and Y.





<u>Hall's Theorem</u> Let $G=(V,E)$ be a bipartite graph with bipartition $V=X\cup Y$. Then

G has a matching covering X $\iff$ $|\text{Neighborhood}(X')| \geq |X'|$ for all $X' \subseteq X$.

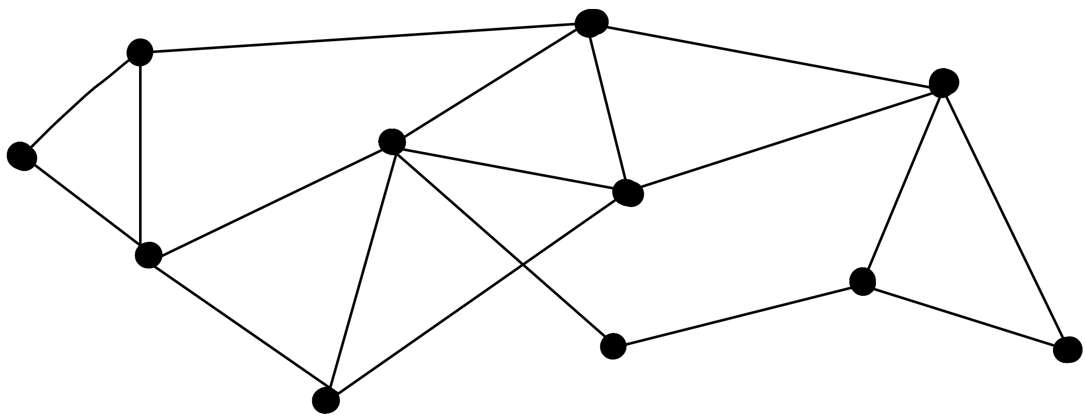


<u>Pf</u> $(\Rightarrow)$ is clear.

We will derive $(\Leftarrow)$ from König's theorem and from the duality of linear programming.

Vertex covers and König's theorem

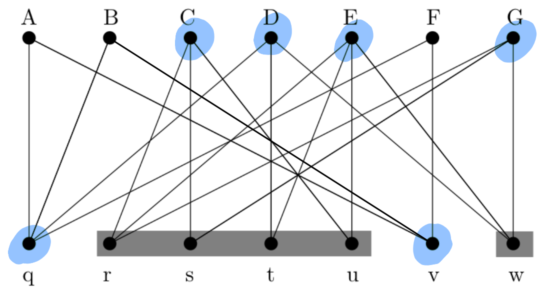
A <u>vertex cover</u> of a graph $G=(V,E)$ is a set $V' \subseteq V$ of vertices so that each edge is incident to at least one vertex in V'.



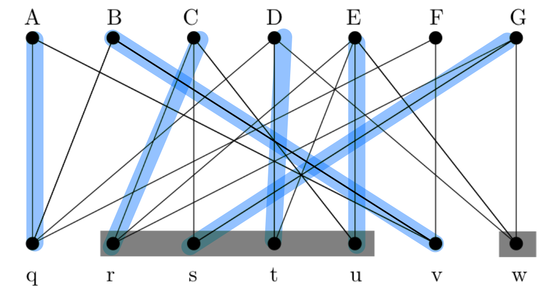
"Dual" to the definition of a matching.

<u>Def</u> A vertex cover is <u>minimum</u> if it has the fewest # of vertices among all vertex covers of G.

<u>König's theorem</u> In a bipartite graph, the size of a minimum vertex cover equals the size of a maximum matching.



Minimum vertex cover



Maximum matching

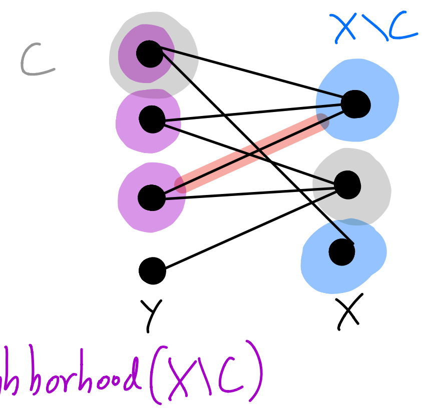
<u>Pf</u> We'll prove this using the duality of linear programming, which furthermore shows how to compute covers/matchings.

Proof of Hall's theorem from König's theorem

Let $G=(V,E)$ be a bipartite graph w/ bipartition $V=X\cup Y$.

If $|\text{Neighborhood}(X')| \geq |X'|$ for all $X' \subseteq X$, we must show G has a matching covering X.

We'll show any vertex cover has size at least $|X|$, which by König's gives a matching of size $|X|$, which obviously covers X.



For a contradiction, suppose there is a vertex cover C with

- k vertices from X
- $< |X|-k$ vertices from Y.

Then $X \setminus C$ has size $|X|-k$ and satisfies $|\text{Neighborhood}(X\setminus C)| \geq |X|-k$ by assumption.

This gives a vertex in $\text{Neighborhood}(X\setminus C)$ that is not in $C\cap Y$, showing C is not a cover.

(The red edge above shows C is not a cover.)

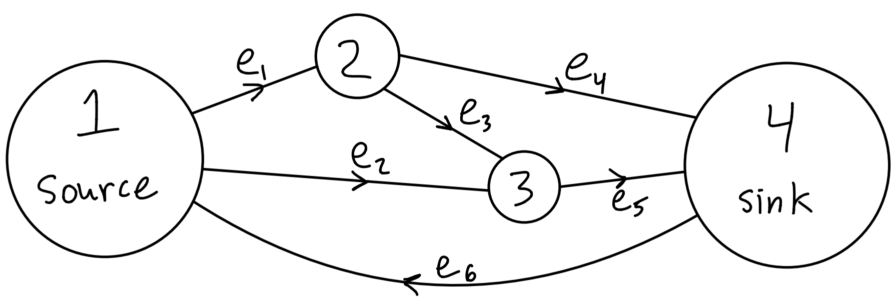
Totally unimodular matrices

A matrix is <u>totally unimodular</u> if every square submatrix has determinant 0, 1, or −1.

Ex

$$
\begin{array}{c c}
 & \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} &
\begin{bmatrix}
-1 & -1 & 0 & 0 & 0 & 1 \\
1 & 0 & -1 & -1 & 0 & 0 \\
0 & 1 & 1 & 0 & -1 & 0 \\
0 & 0 & 0 & 1 & 1 & -1
\end{bmatrix}
\end{array}
$$

In particular, each entry must be 0, −1, or 1.



Non-Ex A matrix of the form is not totally unimodular

$$
\begin{bmatrix}
 & \vdots & & \vdots & \\
\cdots & 1 & \cdots & 1 & \cdots \\
 & \vdots & & \vdots & \\
\cdots & -1 & \cdots & 1 & \cdots \\
 & \vdots & & \vdots &
\end{bmatrix}
$$

Ex If matrix A is totally unimodular, then so is:

$$
\left[\begin{array}{c:c}
 & 0 \\
 & 0 \\
 & 0 \\
A & 0 \\
 & 1 \\
 & 0 \\
 & 0
\end{array}\right]
$$

Thm Consider the linear program

size $m \times n$

$$\text{Max } c^T x \quad \text{subject to} \quad Ax \le b, \quad x \ge 0.$$

If A is totally unimodular, if $b \in \mathbb{Z}^m$, and if there is an optimal solution, then there is an optimal integral solution $x^* \in \mathbb{Z}^n$.

Pf Equational form: $\bar{A}\bar{x} = b$ with $\bar{A} = (A \mid I)$ and $\bar{x} \in \mathbb{R}^{n+m}$.

totally unimodular by above lemma

Solve via simplex method to find an optimal basic feasible solution $\bar{x}^*$ corresponding to basis $B \subseteq \{1, 2, \ldots, n+m\}$ with nonzero entries solving $\bar{x}_B^* = \bar{A}_B^{-1} b$.

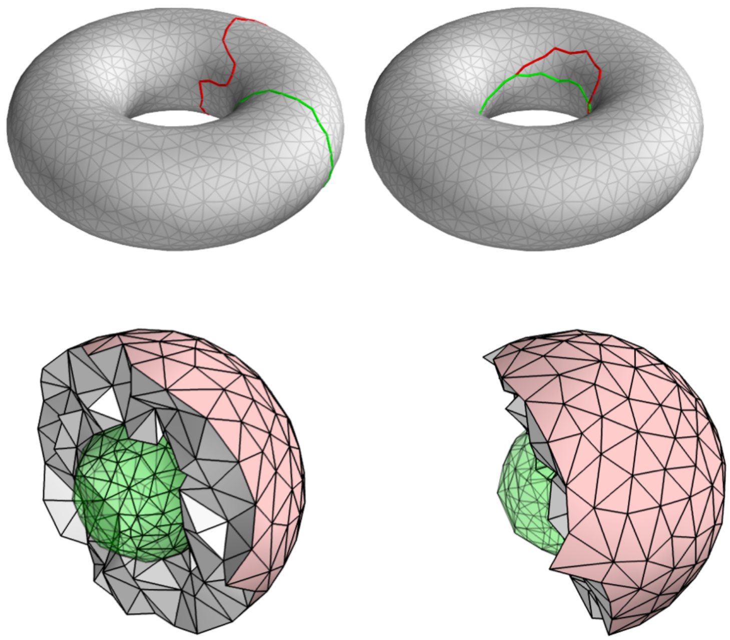
$\bar{A}$ totally unimodular $\Rightarrow \det(\bar{A}_B) \in \{-1, 0, 1\}$.

$\bar{A}_B$ nonsingular $\Rightarrow \det(\bar{A}_B) \in \{-1, 1\}$.

By Cramer's rule, the entries of $\bar{A}_B^{-1}$ are rational numbers with denominator $\det(A_B)$, i.e., integer numbers.

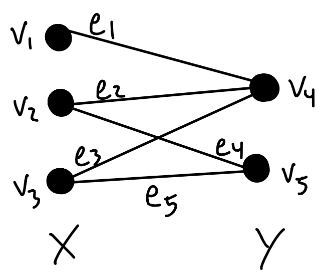
Since $b \in \mathbb{Z}^m$, this gives $\bar{x}^* \in \mathbb{Z}^{n+m}$ and $x^* \in \mathbb{Z}^n$.

"Optimal homologous cycles, total unimodularity, and linear programming"
by Dey, Hirani, and Krishnamoorthy



Total unimodularity and König's theorem

<u>Lemma</u> The incidence matrix A of a bipartite graph $G = (X \cup Y, E)$ is totally unimodular.



$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array} = A$$

Incidence matrix

<u>Pf</u> We must show every $\ell \times \ell$ submatrix Q of A has determinant 0 or ± 1.

<u>Base case $\ell = 1$</u> Clear by def^{n} of incidence matrix.

<u>Inductive step</u> Assume true for $(\ell-1)\times(\ell-1)$ submatrices.

Let Q be an $\ell \times \ell$ submatrix.

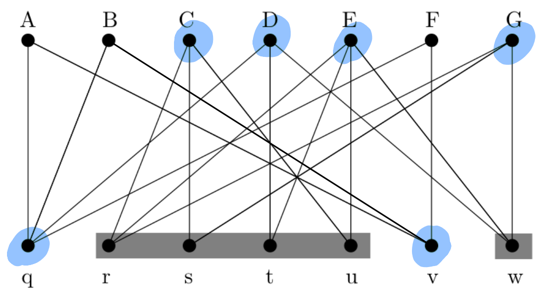
If any column of Q is all zero, determinant is zero.

If any column of Q has one 1, true by induction.

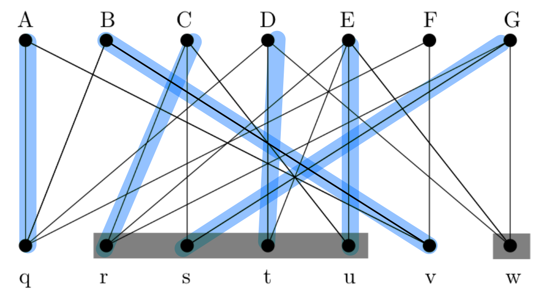
Else all columns of Q have two 1's.

The sum of all rows for vertices in X gives $(1,1,\dots,1)$.
The sum of all rows for vertices in Y gives $(1,1,\dots,1)$.
Hence the rows of Q are linearly dependent
and so $\det(Q) = 0$.

<u>König's theorem</u> In a bipartite graph, the size of a minimum vertex cover equals the size of a maximum matching.



Minimum vertex cover



Maximum matching

<u>Pf</u> Let A be the incidence matrix of the bipartite graph.

Cover: $\text{Min} \sum y_i$

subject to
$$A^T y \geq \vec{1}$$
$$y \geq 0$$
$$y \in \mathbb{Z}^m$$

<u>Optimally</u> $y_i \in \{0,1\}$ for each vertex.

Matching: $\text{Max} \sum x_j$

subject to
$$Ax \leq \vec{1}$$
$$x \geq 0$$
$$x \in \mathbb{Z}^n$$

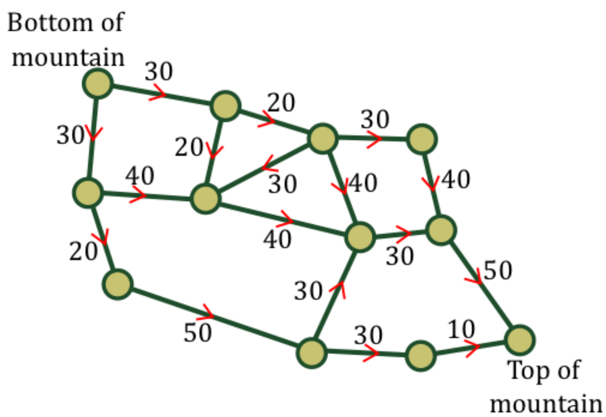
Here $x_j \in \{0,1\}$ for each edge.

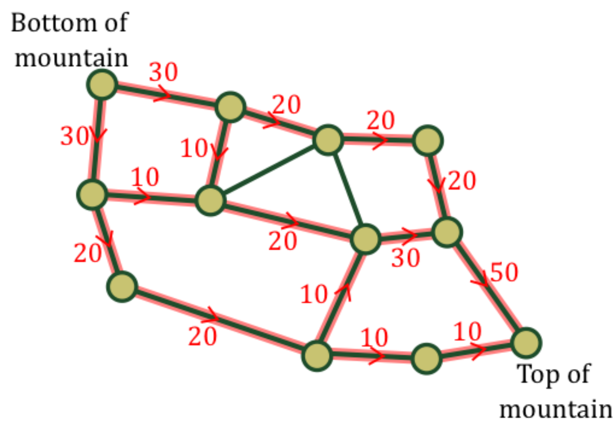
We can drop the integrality constraints since A (and hence A^T) is totally unimodular.

These dual linear programs then have equal optima, proving König's theorem.

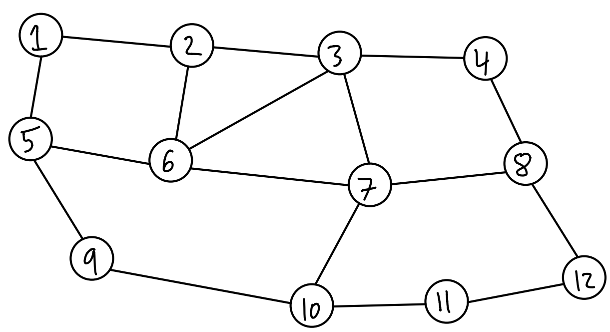
Max flow

How many skiers per minute can ride lifts from the bottom to the top of the mountain?





Images: Kelly Emmrich, Maria Gillespie, Shannon Golden, Rachel Pries



Notation

c_{ij} = capacity of directed edge from i to j.

So $c_{67} = 40$ $\quad c_{76} = 0$

x_{ij} = flow from i to j

So $x_{67} = 20$ $\quad x_{76} = 0$

Let E be the set of directed edges, n = # vertices

Maximize f subject to

$$\sum_{(i,j)\in E} x_{ij} - \sum_{(k,i)\in E} x_{ki} = \begin{cases} f & \text{if } i=1 \\ -f & \text{if } i=n \\ 0 & \text{if } i=2,\dots,n-1 \end{cases}$$

$$x_{ij} \le c_{ij} \quad \text{for all } i,j$$
$$x_{ij} \ge 0$$

We relax to $\le$ without changing the optimum.

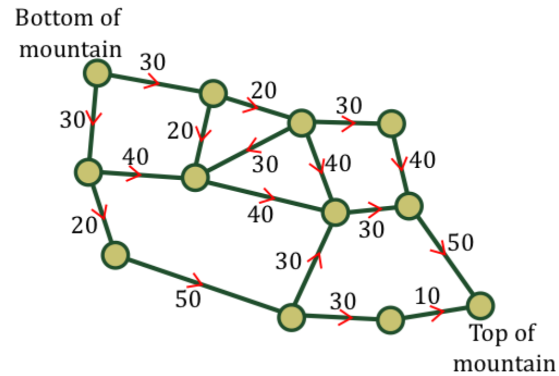
We have: $\quad \text{Max } c^T x$ subject to $Ax \le b$, $x \ge 0$

where

$$x = \begin{bmatrix} f \\ x_{ij} \\ \vdots \end{bmatrix} \quad c = \begin{bmatrix} 1 \\ 0 \\ \vdots \end{bmatrix} \quad A = \begin{array}{c} \\ v_1 \\ \\ \\ \\ v_n \\ \\ \\ x_{ij} \\ \\ \end{array} \begin{array}{c} \qquad\qquad x_{ij} \\ \left[\begin{array}{c|cccc} -1 & & & & \\ 0 & & & & \\ \vdots & & & & \\ 0 & & & & \\ 1 & & & & \\ \hline 0 & 1 & & & \\ & & 1 & & \\ \vdots & & & \ddots & \\ 0 & & & & 1 \end{array}\right] \end{array} \quad b = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \hline c_{ij} \end{bmatrix}$$

1 and -1 in each column

Min cut



The dual of the max flow problem is

$$\text{Min } b^T y \quad \text{subject to} \quad A^T y \geq c, \quad y \geq 0$$

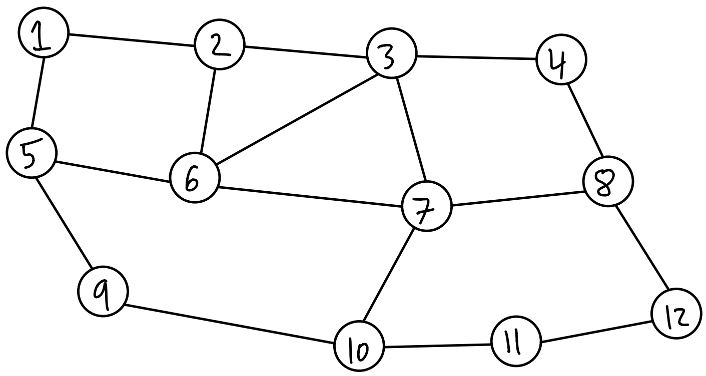
where

$$y = \begin{bmatrix} u_1 \\ \vdots \\ u_{12} \\ \hline \\ y_{ij} \\ \\ \end{bmatrix} \qquad b = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \hline \\ c_{ij} \\ \\ \end{bmatrix} \qquad A^T = \left[\begin{array}{c|cccc} -1\ 0 \cdots 0\ 1 & 0 & & \cdots & 0 \\ \hline & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \\ & & & & & 1 \end{array}\right] \qquad c = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ \\ \end{bmatrix}$$

1 and −1 in each row

In equations, this is $\displaystyle \text{Min} \sum_{(i,j)\in E} c_{ij}\, y_{ij}$

subject to
$$-u_1 + u_n \geq 1$$
$$u_i - u_j + y_{ij} \geq 0 \quad \text{for all } (i,j) \in E$$
$$u_i \geq 0 \qquad u_i \in \mathbb{Z}$$
$$y_{ij} \geq 0 \qquad y_{ij} \in \mathbb{Z}$$



We'll see A^T is totally unimodular.
So since c is integral, we can add the constraints $u_i \in \mathbb{Z}$, $y_{ij} \in \mathbb{Z}$.

In an optimal solution:

- can take $u_i \in \{0,1\}$
- $(u_i, u_j) = (0,0), (1,1)$ or $(1,0)$ implies $y_{ij} = 0$
- $(u_i, u_j) = (0,1)$ implies $y_{ij} = 1$

We get:

$$\text{Min} \sum_{(i,j)\in E} c_{ij} y_{ij}$$

subject to

$$-u_1 + u_n \geq 1$$
$$u_i - u_j + y_{ij} \geq 0 \quad \text{for all } (i,j) \in E$$
$$u_i \in \{0,1\}$$
$$y_{ij} = \{0,1\}$$

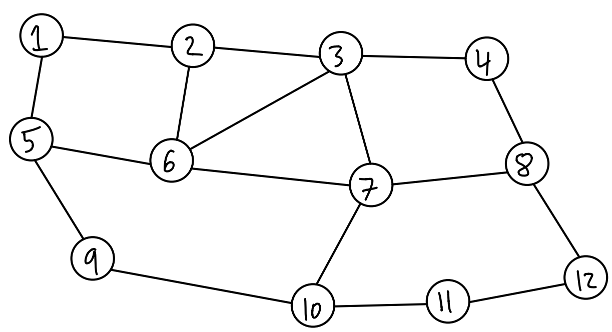
This cuts the network into two sets:

V_0 = {vertices with $u_i = 0$, including source}.

V_1 = {vertices with $u_i = 1$, including sink}.

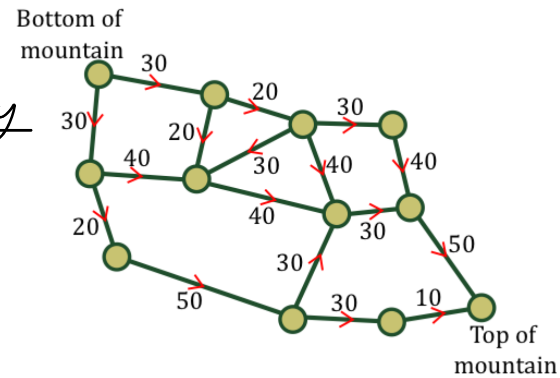
$$y_{ij} = \begin{cases} 1 & \text{if } i \in V_0,\ j \in V_1 \\ 0 & \text{otherwise} \end{cases}$$

Hence the dual of max flow is min cut!



Min cut and total unimodularity

$$\text{Min} \sum_{(i,j)\in E} c_{ij} y_{ij} \quad \text{subject to}$$

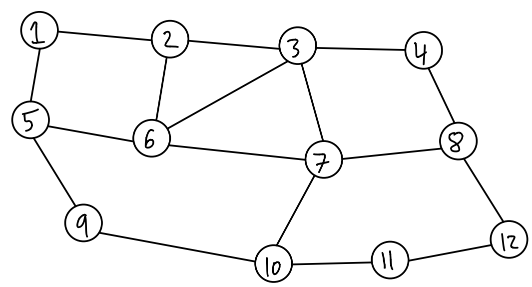


$$-u_1 + u_n \geq 1$$

$$u_i - u_j + y_{ij} \geq 0 \quad \text{for all } (i,j)\in E$$

$$u_i \geq 0 \qquad u_i \in \mathbb{Z}$$

$$y_{ij} \geq 0 \qquad y_{ij} \in \mathbb{Z}$$



$$\text{Min} \quad b^T y \quad \text{subject to} \quad A^T y \geq c, \; y \geq 0$$

$$y = \begin{bmatrix} u_1 \\ \vdots \\ u_{12} \\ \hline \\ y_{ij} \\ \\ \end{bmatrix} \qquad b = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \hline \\ c_{ij} \\ \\ \end{bmatrix} \qquad A^T = \begin{array}{c} \\ \\ ij \\ \\ \end{array} \begin{array}{c} \begin{array}{cc} u_i & \quad y_{ij} \end{array} \\ \left[\begin{array}{cccc|cccc} -1 & 0 & \cdots & 0 \; 1 & 0 & & \cdots & 0 \\ \hline & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{array}\right] \end{array} \qquad c = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ \\ \end{bmatrix}$$

1 and -1 in each row

<u>Lemma</u> The matrix A^T is totally unimodular.

<u>Pf</u> We must show every $\ell \times \ell$ submatrix Q of A^T has determinant 0 or ± 1.

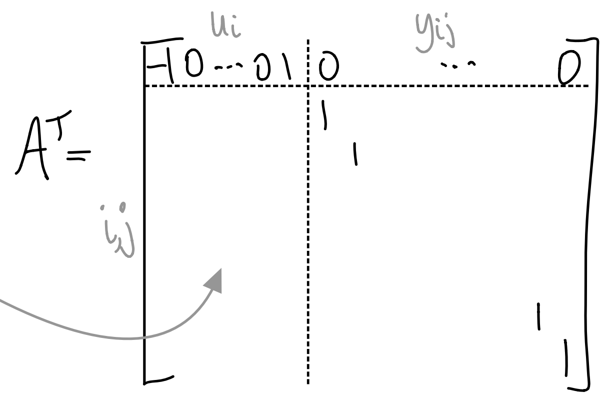
<u>Base case $\ell = 1$</u> Clear.

<u>Inductive step</u> Assume true for $(\ell-1)\times(\ell-1)$ submatrices.

Let Q be an $\ell \times \ell$ submatrix.

If any column of Q is from RHS, true by induction.

1 and -1 in each row



Else all columns of Q are from LHS.

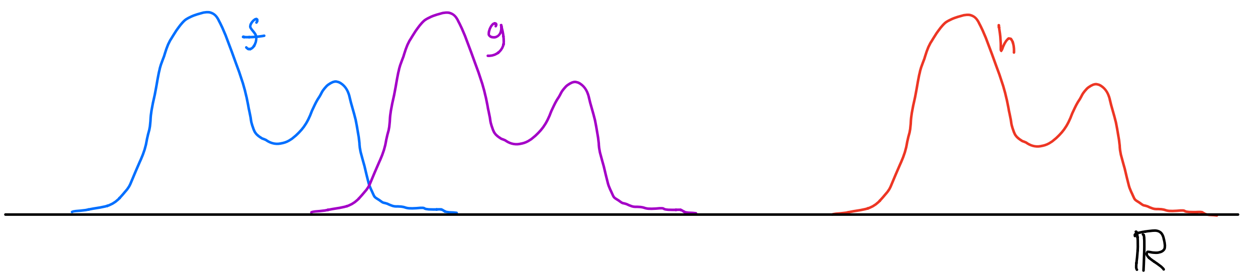
If any row has at most one ± 1, true by induction.

Else all rows have a 1 and -1.

Summing the columns gives the zero vector, so $\det(Q) = 0$.

Introduction to optimal transport

(Also called the Wasserstein, Kantorovich-Rubinstein, or earth mover's distance.)

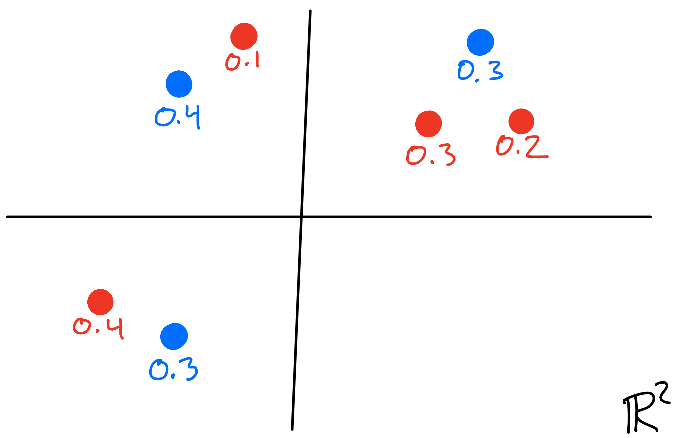


Using function distances, $\|f-g\|_\infty \approx \|f-h\|_\infty$.

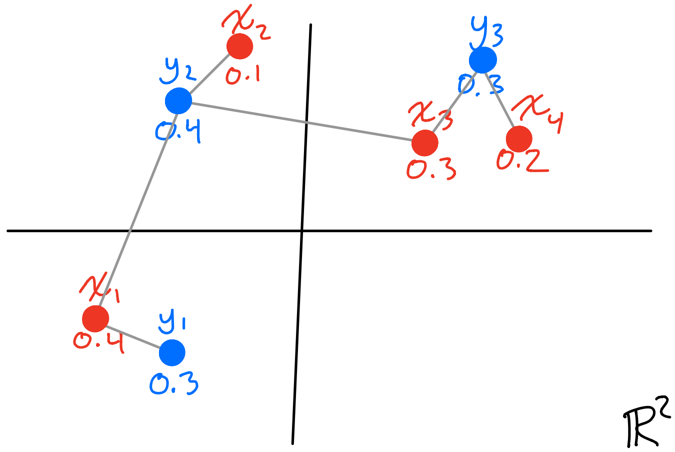
Using the Wasserstein distance, $d_W(f,g) < d_W(f,h)$.

What about geodesics?

Introduction to optimal transport



Introduction to optimal transport

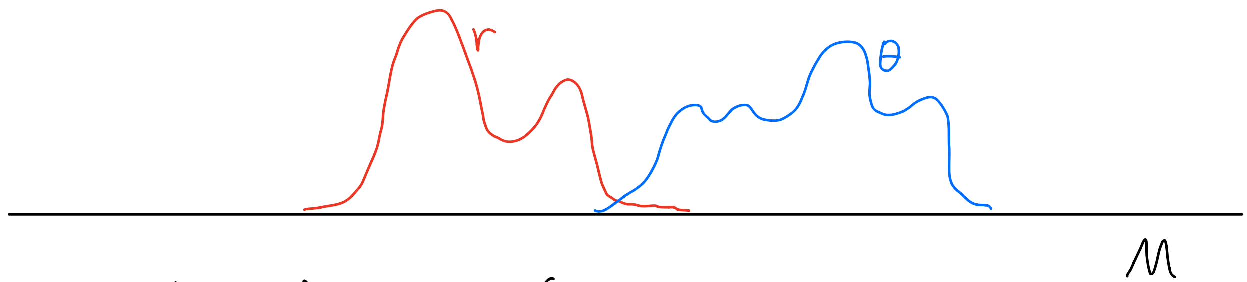


		x_1 0.4	x_2 0.1	x_3 0.3	x_4 0.2
y_1	0.3	0.3	0	0	0
y_2	0.4	0.1	0.1	0.2	0
y_3	0.3	0	0	0.1	0.2

$$d_W\left(\sum_i \alpha_i \delta_{x_i}, \sum_j \beta_j \delta_{y_j}\right)$$

$$= \min\left\{\sum_{i,j} \gamma_{i,j}\, d(x_i, y_j) : \gamma_{i,j} \geq 0, \ \sum_i \gamma_{i,j} = \beta_j, \ \sum_j \gamma_{i,j} = \alpha_i\right\}$$

Introduction to optimal transport



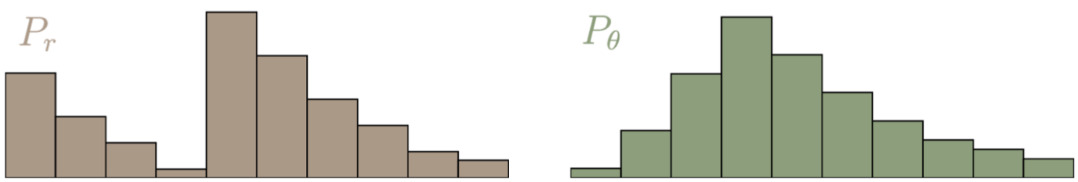
$$d_{W_1}(r, \theta) = \inf_{\gamma} \int_{M \times M} d(x,y)\, d\gamma(x,y)$$

where γ is a measure on $M \times M$ with marginals r and θ.

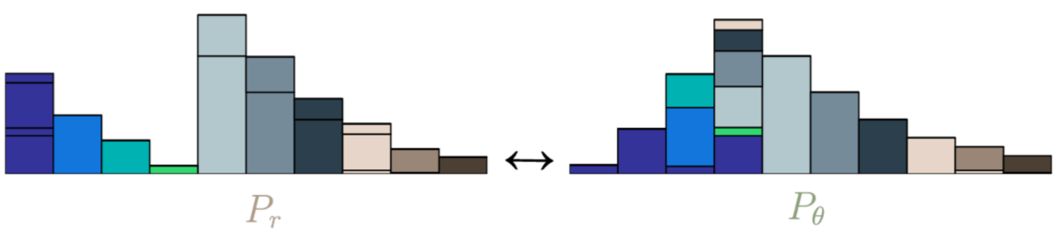
Introduction to optimal transport

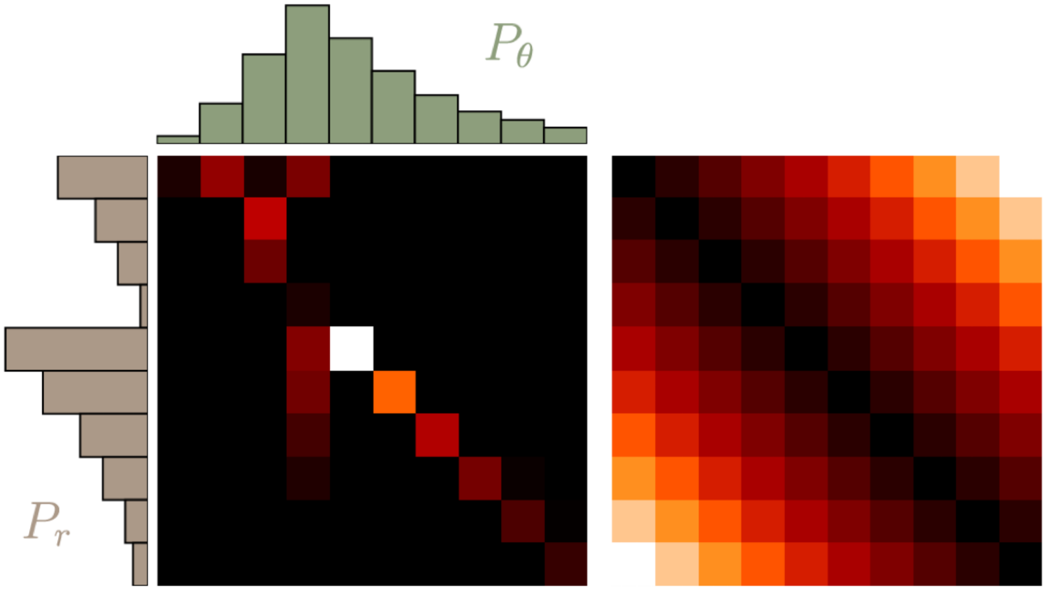
See vincentherrmann.github.io/blog/wasserstein
for the following images and ideas!

Connection to GANS:



Let r and θ be two measures on the same space.





Transport plan γ Distances

Vectorize to get x Vectorize to get c

Minimize $c^T x$ subject to $Ax=b$, $x \geq 0$.

$$\left.\left[\begin{array}{cccc|cccc} P_r(x_1) & P_r(x_2) & \dots & P_r(x_n) & P_\theta(y_1) & P_\theta(y_2) & \dots & P_\theta(y_n) \end{array}\right]\right\} \mathbf{b}^T$$

$$\mathbf{x}\left\{\begin{bmatrix} \gamma(x_1,y_1) \\ \gamma(x_1,y_2) \\ \vdots \\ \hline \gamma(x_2,y_1) \\ \gamma(x_2,y_2) \\ \vdots \\ \hline \vdots \\ \hline \gamma(x_n,y_1) \\ \gamma(x_n,y_2) \\ \vdots \end{bmatrix}\right. \quad \left.\left[\begin{array}{cccc|cccc} 1 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 1 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline 0 & 0 & \dots & 1 & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \ddots & \vdots \end{array}\right]\right\} \mathbf{A}^T$$

Areas for improvement:

- If r and θ have supports of size n, then our linear program has n^2 variables.

- We can find the Wasserstein distance without finding a transport plan.

- Wasserstein GANs needs to compute a gradient with respect to θ, which here is a constraint.

Kantorovich – Rubinstein Duality

Wasserstein distance:

$$d_W(r, \theta) = \inf_{\gamma} \int_{M \times M} d(x,y) \, d\gamma(x,y)$$

where γ is a measure on $M \times M$ with marginals r and θ

$$= \sup \left\{ \int_M f(x)\, dr - \int_M f(x)\, d\theta \;\middle|\; f: M \to \mathbb{R} \text{ is 1-Lipschitz} \right\}.$$

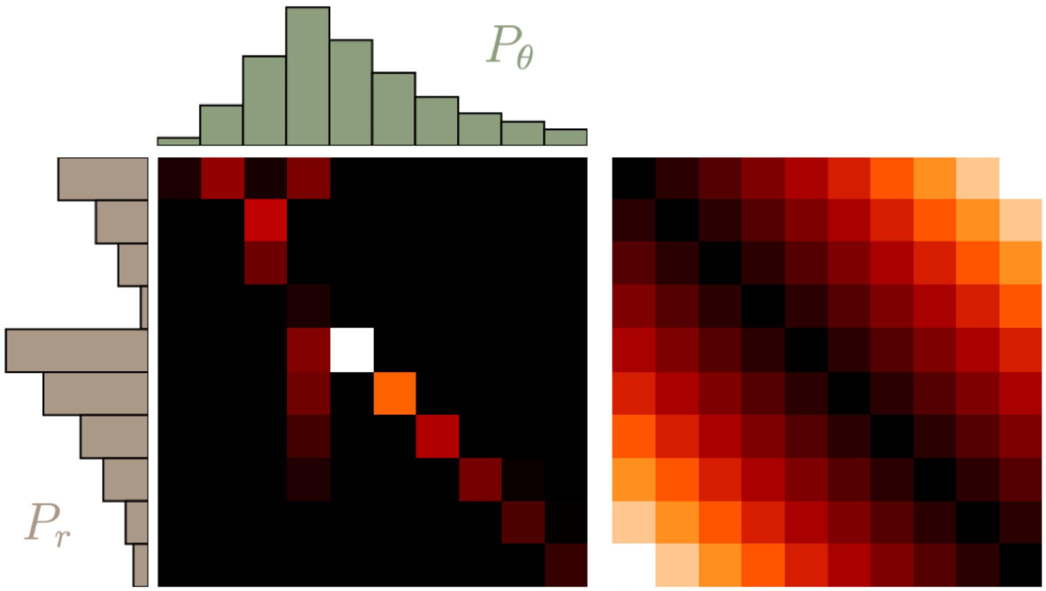
means $|f(x) - f(x')| \leq d(x,x')$

As a linear program:

Primal: Minimize $c^T x$ subject to $Ax = b$, $x \geq 0$.

Dual: Maximize $b^T y$ subject to $A^T y \leq c$.

Primal: Minimize $c^T x$ subject to $Ax=b$, $x \geq 0$.



Transport plan γ

Vectorize to get x

Distances

Vectorize to get c

$$\left.\begin{bmatrix} P_r(x_1) & P_r(x_2) & \dots & P_r(x_n) & P_\theta(y_1) & P_\theta(y_2) & \dots & P_\theta(y_n) \end{bmatrix}\right\} \mathbf{b}^T$$

$$\mathbf{x}\left\{\left[\begin{array}{c} \gamma(x_1, y_1) \\ \gamma(x_1, y_2) \\ \vdots \\ \hline \gamma(x_2, y_1) \\ \gamma(x_2, y_2) \\ \vdots \\ \hline \vdots \\ \hline \gamma(x_n, y_1) \\ \gamma(x_n, y_2) \\ \vdots \end{array}\right]\right. \quad \left.\left[\begin{array}{cccc|cccc} 1 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 1 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline 0 & 0 & \dots & 1 & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \ddots & \vdots \end{array}\right]\right\} \mathbf{A}^T$$

Dual: Maximize $b^T y$ subject to $A^T y \leq c$.

$$
\underbrace{\left[\begin{array}{ccc|ccc|c|ccc} \mathbf{D}_{1,1} & \mathbf{D}_{1,2} & \dots & \mathbf{D}_{2,1} & \mathbf{D}_{2,2} & \dots & \dots & \mathbf{D}_{n,1} & \mathbf{D}_{n,2} & \dots \end{array}\right]}_{\mathbf{c}^T}
$$

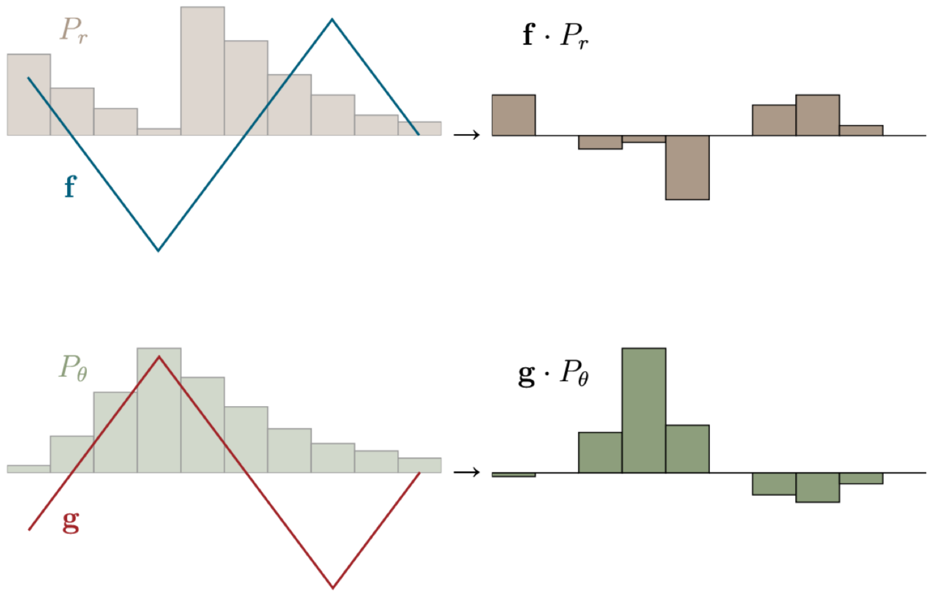
$$
\underbrace{\left[\begin{array}{c} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \\ \hline g(x_1) \\ g(x_2) \\ \vdots \\ g(x_n) \end{array}\right]}_{\mathbf{y}}
\qquad
\underbrace{\left[\begin{array}{ccc|ccc|c|ccc}
1 & 1 & \dots & 0 & 0 & \dots & \dots & 0 & 0 & \dots \\
0 & 0 & \dots & 1 & 1 & \dots & \dots & 0 & 0 & \dots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\
0 & 0 & \dots & 0 & 0 & \dots & \dots & 1 & 1 & \dots \\
\hline
1 & 0 & \dots & 1 & 0 & \dots & \dots & 1 & 0 & \dots \\
0 & 1 & \dots & 0 & 1 & \dots & \dots & 0 & 1 & \dots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \dots & \vdots & \vdots & \ddots \\
0 & 0 & \dots & 0 & 0 & \dots & \dots & 0 & 0 & \dots
\end{array}\right]}_{\mathbf{A}}
$$

Constraints: $f(x_i) + g(x_j) \leq D_{i,j} = d(x_i, x_j)$

Taking $i=j$ gives $f(x_i) + g(x_i) \leq 0$

Optimally, equality occurs, giving $g(x_i) = -f(x_i)$
or $g = -f$.

So $|f(x_i) - f(x_j)| \leq d(x_i, x_j)$ gives f is 1-Lipschitz.



$$\text{Max } b^T y = f_r(x_1)P_r(x_1) + \dots + f_r(x_n)P_r(x_n) + \text{(same for } g \text{ and } \theta)$$
$$= \int f(x)\,dr + \int g(x)\,d\theta$$
$$= \int f(x)\,dr - \int f(x)\,d\theta$$

Kantorovich – Rubinstein Duality

$$d_W(r,\theta) = \inf_{\gamma} \int_{M\times M} d(x,y)\, d\gamma(x,y)$$

where γ is a measure on $M\times M$ with marginals r and θ

$$= \sup\left\{ \int_M f(x)\,dr - \int_M f(x)\,d\theta \;\middle|\; f: M\to\mathbb{R} \text{ is 1-Lipschitz} \right\}.$$

means $|f(x)-f(x')| \le d(x,x')$

Improvements:

- If r and θ have supports of size n, then our linear program has n^2 variables. Dual has only n variables.
- We can find the Wasserstein distance without finding a transport plan using dual.
- Wasserstein GANs needs to compute a gradient with respect to θ, which ... is possible with the dual!

Title: Neighborly polytopes and the sparsity-promoting ℓ^1 norm

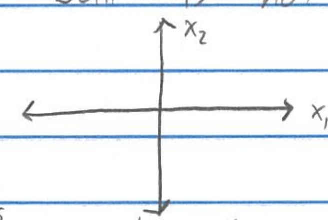
References: Donoho technical report, 2005
Donoho & Tanner, PNAS, 2005

Fix $A \in \mathbb{R}^{d \times n}$, $y \in \mathbb{R}^d$ with $d < n$. Let $x \in \mathbb{R}^n$.

(0) $\min \|x\|_0$ subject to $y = Ax$

underdetermined

This is NP-hard ; the ℓ^0 ball is not convex

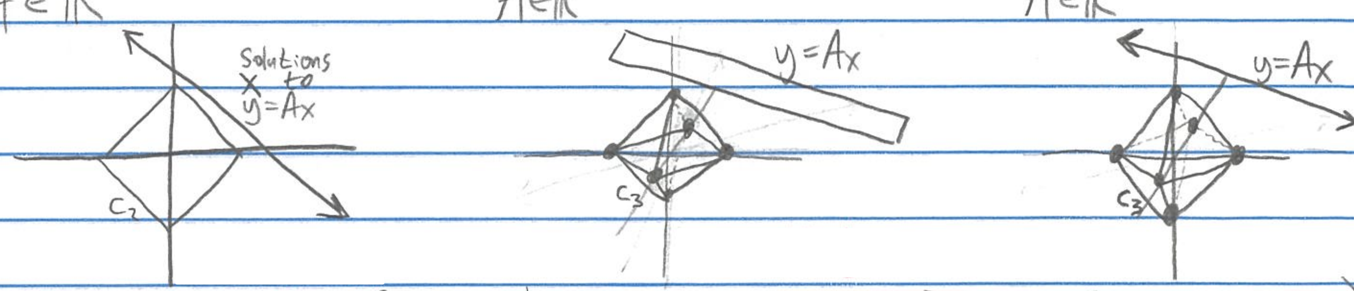


$\{x \in \mathbb{R}^n \mid \|x\|_0 \le 1\}$

Consider the convex relaxation

(1) $\min \|x\|_1$ subject to $y = Ax$

Pics $A \in \mathbb{R}^{1\times 2}$ $A \in \mathbb{R}^{1\times 3}$ $A \in \mathbb{R}^{2\times 3}$



Cross-polytope $C_n = \{x \in \mathbb{R}^n \mid \|x\|_1 = |x_1| + \dots + |x_n| \le 1\} = \mathrm{Conv}(\{\pm e_1, \dots, \pm e_n\})$

"Equivalence" of ℓ^0 and ℓ^1 optimization

Important Corollary:

The overwhelming majority of $A \in \mathbb{R}^{n \times d}$, $y \in \mathbb{R}^d$
(A a random orthogonal projection)
with n large (larger than pictures on prior page)
and $d \leq \lfloor 0.7\, n \rfloor$ have the property that
if x is a solution to (0) with less than
$0.49d$ nonzeros, then x is also the
unique solution to (1).

Main Theorem $A \in \mathbb{R}^{d \times n}$ with $d < n$. Then

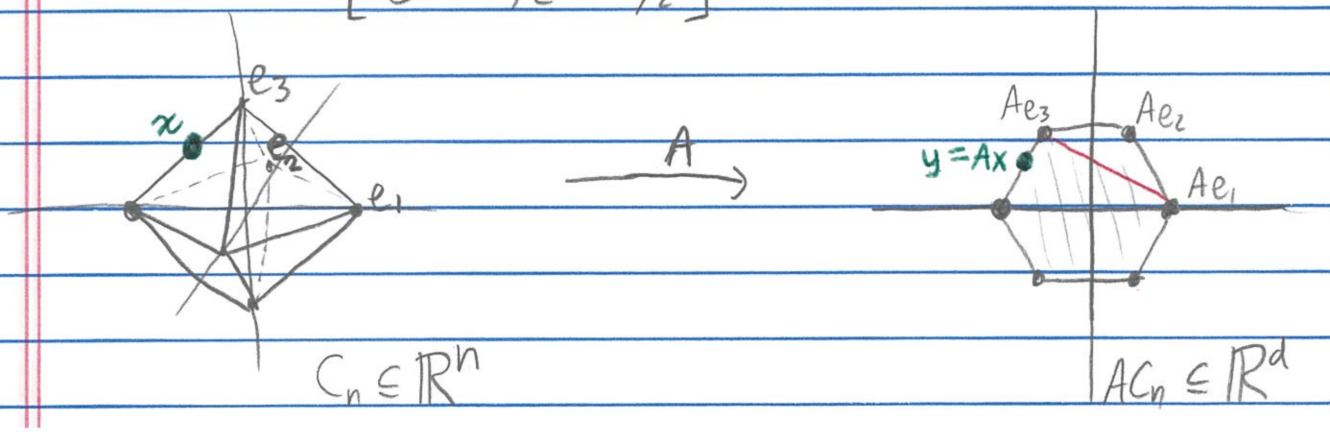
- the polytope AC_n has $2n$ vertices and is k-neighborly

$\Longleftrightarrow$

- whenever $y = Ax$ has a solution x with at most $k+1$ nonzeros, x is the unique solution to (1).

Here $AC_n = \mathrm{Conv}(\{\pm Ae_1, \ldots, \pm Ae_n\}) = \mathrm{Conv}(\{\pm \text{ each column of } A\})$

Ex $A = \begin{bmatrix} 1 & 1/2 & -1/2 \\ 0 & \sqrt{3}/2 & \sqrt{3}/2 \end{bmatrix}$



Note C_n and AC_n are <u>centrally symmetric polytopes</u>, meaning $C_n = -C_n$ and $AC_n = -AC_n$, i.e. reflecting through the origin leaves them unchanged.

<u>Def</u> A centrally symmetric polytope is <u>k-neighborly</u> if any collection of $k+1$ vertices not including an antipodal pair form a face.

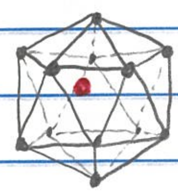
<u>Ex</u> The cross-polytope C_n in $\mathbb{R}^n$ is $(n-1)$-neighborly.

<u>Ex</u> AC_n will often be k-neighborly for k relatively large, especially if A is a special matrix (Fourier, partial Vandermonde, augmented Hadamard, incoherent dictionary, signal processing, error correcting codes).

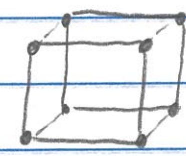
(The corollary compares the expected # of faces of AC_n to those of C_n.)

Proof of ($\Longrightarrow$) in Main Theorem

Fact k-neighborly polytopes for $k \geq 1$ are simplicial
(all faces are simplices).



Simplicial



Not simplicial

Lemma 1 If y is a point in a face of a simplicial polytope, then y has a unique representation as a convex combination of vertices, which all belong to the face.

Lemma 2 If AC_n has $2n$ vertices and is k-neighborly,
then F is an i-face of C_n
$\Longleftrightarrow$ AF is an i-face of AC_n
for all $0 \leq i \leq k$

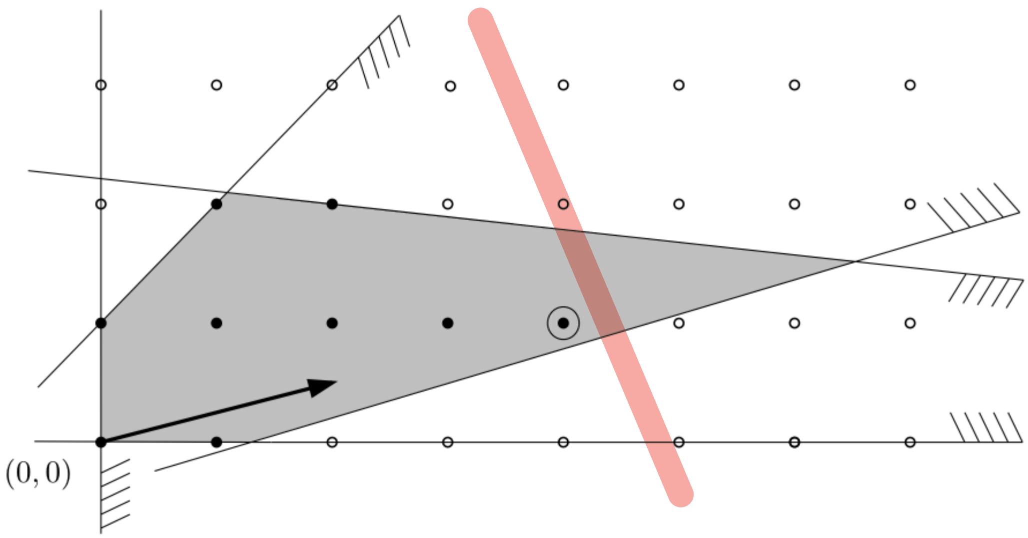
Proof of ($\Longrightarrow$) in Main Theorem

Suppose $x \in \mathbb{R}^n$ has at most $k+1$ nonzeros.
So x is in a k-face of a scaled C_n.
By Lemma 2, $y = Ax$ is in a k-face of a scaled AC_n;
so x is a solution to (1).
Furthermore, by Lemma 1 x is the unique solution to (1).
(So (1) "magically" finds the solution to (0)!)

Cutting planes

Maximize $c^T x$

subject to $Ax \leq b$, $x \geq 0$, $x \in \mathbb{Z}^n$



The Gomory fractional cut

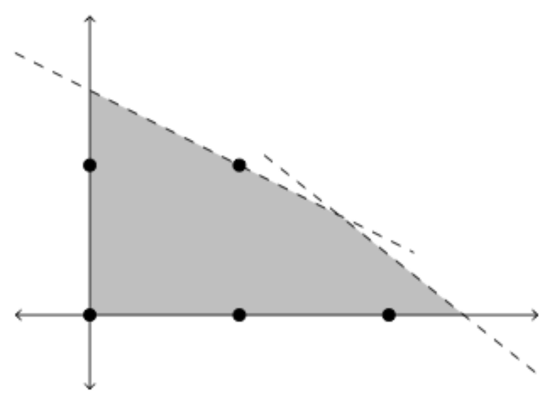
These images and content are from Mikhail Lavrov, Math 482:
faculty.math.illinois.edu/~mlavrov/docs/482-fall-2019/lecture35.pdf

Maximize $c^T x$
subject to $Ax \le b, \quad x \ge 0, \quad x \in \mathbb{Z}^n$

Assume all entries in A and b are integers
(Easy to obtain if they're all rationals.)

Ex Maximize $2x_1 + 3x_2$

subject to
$$x_1 + 2x_2 \le 3$$
$$4x_1 + 5x_2 \le 10$$
$$x_1, x_2 \ge 0$$
$$x_1, x_2 \in \mathbb{Z}$$



Equational form Max $2x_1 + 3x_2$

subject to
$$x_1 + 2x_2 + x_3 \qquad = 3$$
$$4x_1 + 5x_2 \qquad + x_4 = 10$$
$$x_1, x_2, x_3, x_4 \ge 0$$
$$x_1, x_2, x_3, x_4 \in \mathbb{Z}$$

Simplex tableau

$$\begin{aligned} x_3 &= 3 - x_1 - 2x_2 \\ x_4 &= 10 - 4x_1 - 5x_2 \\ \hline z &= 2x_1 + 3x_2 \end{aligned} \xrightarrow{x_2} \begin{aligned} x_2 &= \tfrac{3}{2} - \tfrac{1}{2}x_1 - \tfrac{1}{2}x_3 \\ x_4 &= \tfrac{5}{2} - \tfrac{3}{2}x_1 + \tfrac{5}{2}x_3 \\ \hline z &= \tfrac{9}{2} + \tfrac{1}{2}x_1 - \tfrac{3}{2}x_3 \end{aligned}$$

$$\xrightarrow{x_1} \begin{aligned} x_1 &= \tfrac{5}{3} + \tfrac{5}{3}x_3 - \tfrac{2}{3}x_4 \\ x_2 &= \tfrac{2}{3} - \tfrac{4}{3}x_3 + \tfrac{1}{3}x_4 \\ \hline z &= \tfrac{16}{3} - \tfrac{2}{3}x_3 - \tfrac{1}{3}x_4 \end{aligned}$$

Not integers; pick either row! We'll use the 2nd:

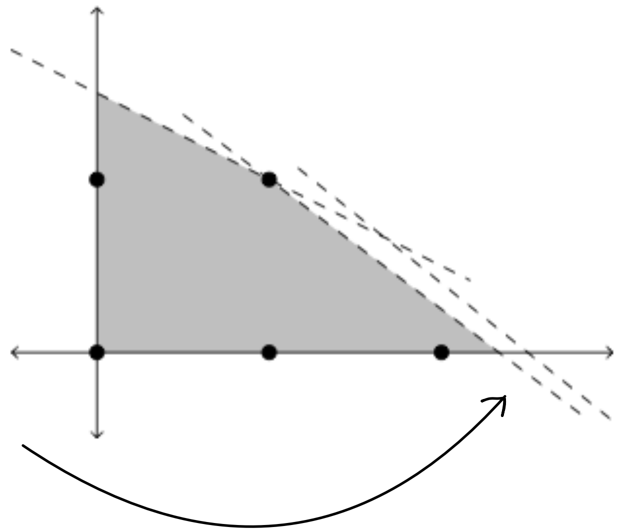
$$x_2 + \frac{4}{3}x_3 - \frac{1}{3}x_4 = \frac{2}{3}$$

$$\underbrace{x_2 + x_3 - x_4}_{\text{integer constants and variables}} + \underbrace{\frac{1}{3}x_3 + \frac{2}{3}x_4}_{\text{nonnegative constants and variables}} = \frac{2}{3}$$

Drop nonnegative part to get $x_2 + x_3 - x_4 \le \frac{2}{3}$
which since all variables are integers gives $x_2 + x_3 - x_4 \le 0$.

This is Gomory's cut, the extra constraint we'll add to our problem.

Visualization



$$x_2 + x_3 - x_4$$
$$= x_2 + (3 - x_1 - 2x_2) - (10 - 4x_1 - 5x_2)$$
$$= -7 + 3x_1 + 4x_2$$

So we have $3x_1 + 4x_2 \leq 7$

More generally For nonnegative integer variables $x_1, \ldots, x_n$,

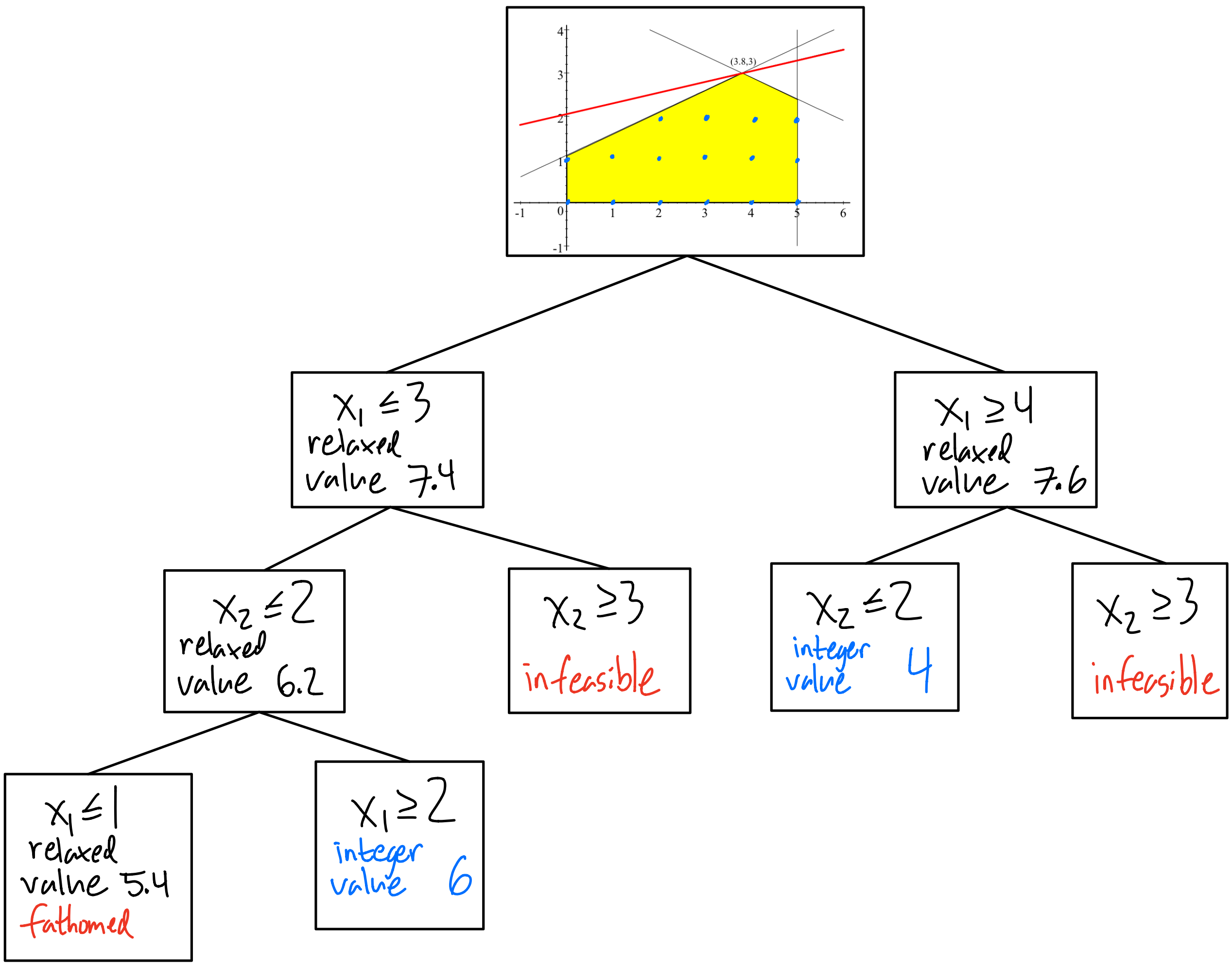
$$a_1 x_1 + \ldots + a_n x_n = b$$

implies

$$\lfloor a_1 \rfloor x_1 + \ldots + \lfloor a_n \rfloor x_n \leq \lfloor b \rfloor.$$

Proof

Branch and bound

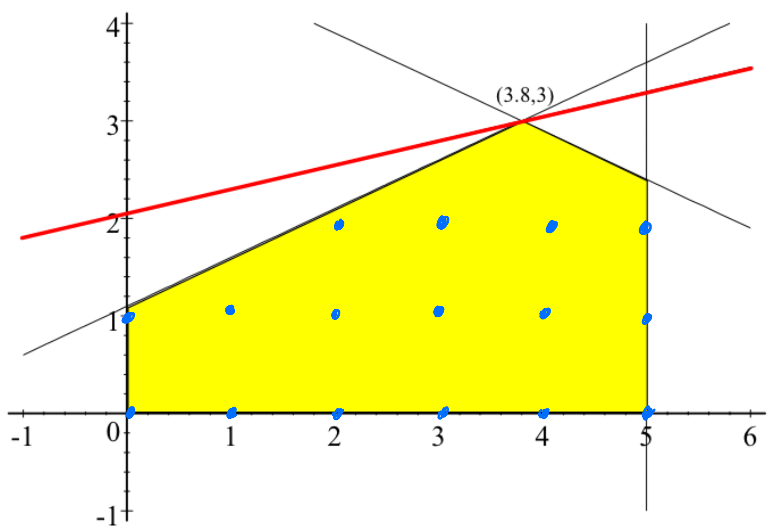


Branch and bound

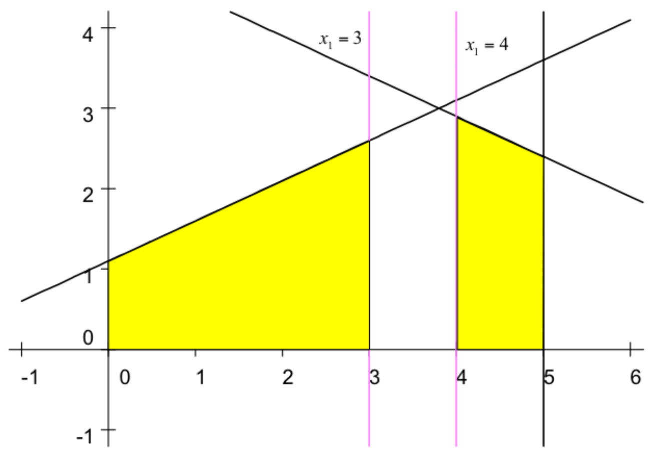
These images and content are from Mustafa Çelebi Pinar: ie.bilkent.edu.tr/~mustafap/courses/bb.pdf.

Example Maximize $-x_1 + 4x_2$

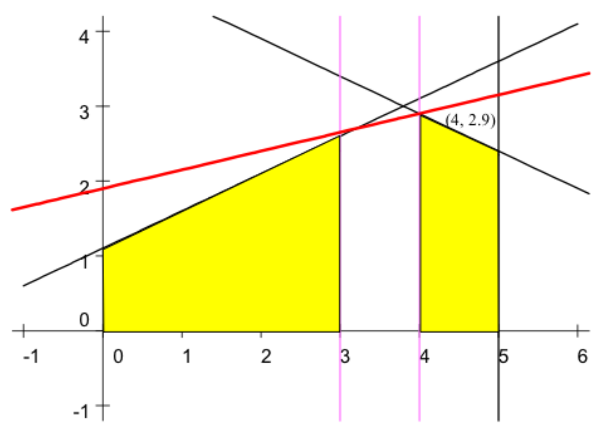
subject to
$$-10x_1 + 20x_2 \le 22$$
$$5x_1 + 10x_2 \le 49$$
$$x_1 \le 5$$
$$x_1, x_2 \ge 0$$
$$x_1, x_2 \in \mathbb{Z}$$



Branch: Since $x_1 \in \mathbb{Z}$, either $x_1 \geq 4$ or $x_1 \leq 3$.



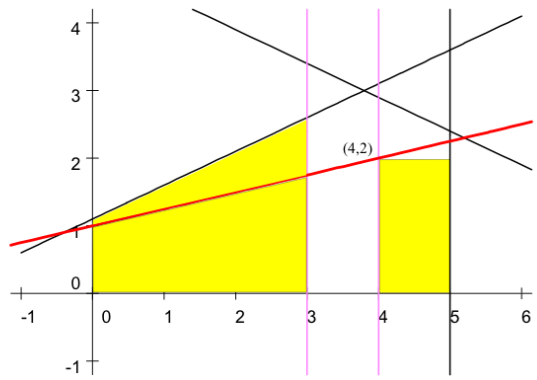
Let's first explore the right side $x_1 \geq 4$:
Relaxation has non-integer solution $(4, 2.9)$ with value 7.6



Branch: Since $x_2 \in \mathbb{Z}$, either $x_2 \geq 3$ or $x_2 \leq 2$.

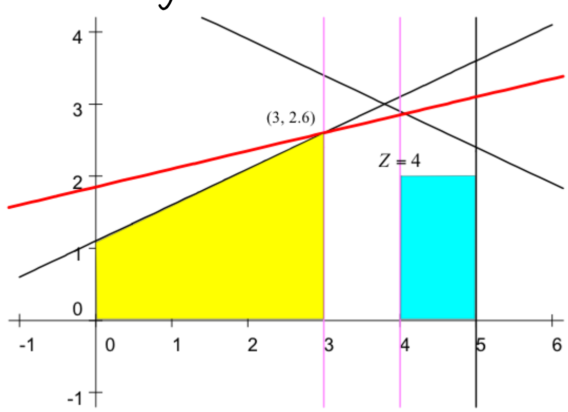
$x_2 \geq 3$ gives an infeasible relaxation.

$x_2 \le 2$: relaxation has integer solution $(4,2)$ with value 4.



This is our current best value for the integer problem.

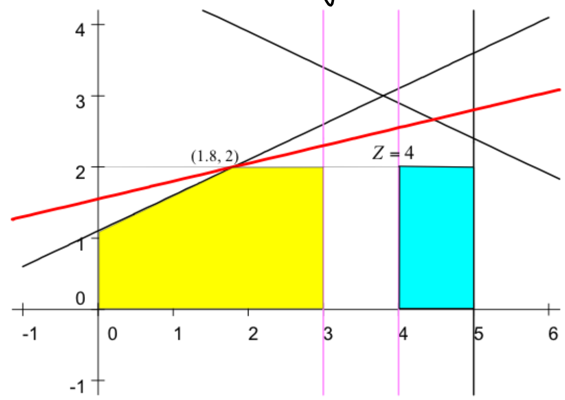
We now explore the left side $x_1 \le 3$:
Relaxation has non-integer solution $(3, 2.6)$ with value 7.4.



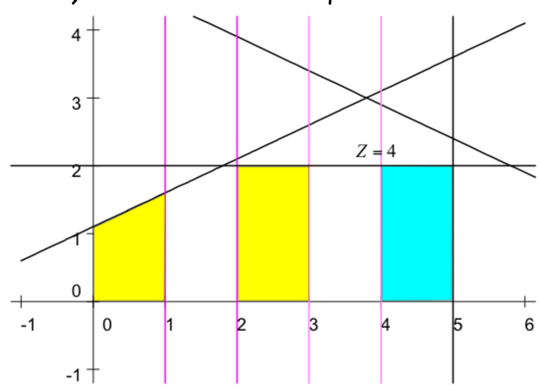
Branch: Since $x_2 \in \mathbb{Z}$, either $x_2 \ge 3$ or $x_2 \le 2$.

$x_2 \ge 3$ gives an infeasible relaxation.

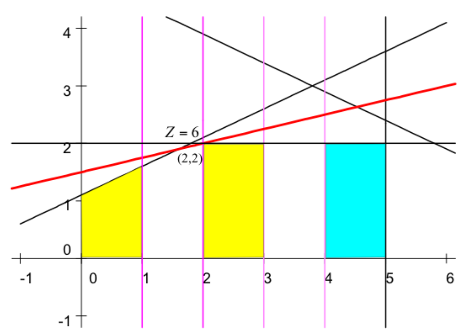
$x_2 \leq 2$: Relaxation has non-integer solution $(1.8, 2)$ with value 6.2



Branch: Since $x_1 \in \mathbb{Z}$, either $x_1 \geq 2$ or $x_1 \leq 1$.

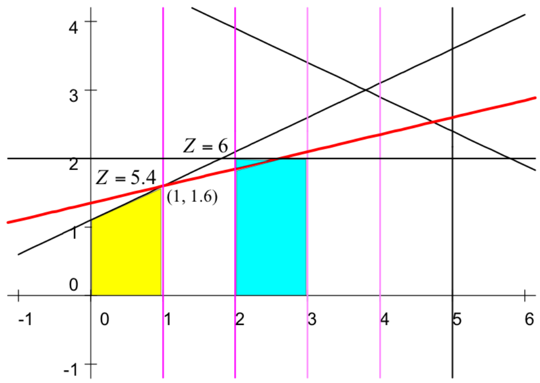


$x_1 \geq 2$: Relaxation has integer solution $(2,2)$ with value 6.



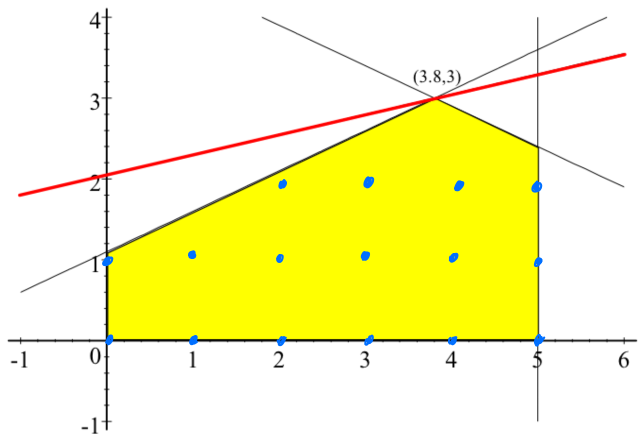
This is our current best value for the integer problem.

$x_1 \le 1$: Relaxation has non-integer solution $(1, 1.6)$ with value 5.4.



Since $5.4 \le 6$, the rest of this branch can be ignored — it is "fathomed".

Hence the optimal solution is $(2,2)$ with value 6.



Here is the tree we followed: