Unit 1: Linear Programming

Outline

1. Definitions and Standard Form

A "Linear Program" (LP) is given by minimizing/maximizing a linear function, denoted $c^T x = \sum_{i=1}^{n} c_i x_i$ or $\langle c, x \rangle$, over a feasible region given by a finite collection of linear inequalities, denoted $a_i^T x \le b_i$ for $i = 1 \ldots m$.

Written concisely, for $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$,

$$\begin{cases} \min/\max & c^T x \\ \text{s.t.} & Ax \le b \end{cases}$$

elementwise

Notationally, we will have a_i^T denote a row of A
and A_i denote a column of A as

$$A = \begin{bmatrix} — a_1^T — \\ \vdots \\ — a_m^T — \end{bmatrix} = \begin{bmatrix} | & & | \\ A_1 & \cdots & A_n \\ | & & | \end{bmatrix}$$

Notes

(1) This model implicitly allows linear equality constraints

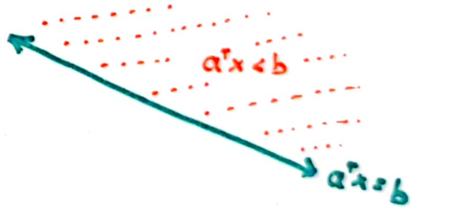
as $a_i^T x = b$ if and only if $a_i^T x \leq b_i$
and $-a_i^T x \leq -b_i$.

(2) This model implicitly allows nonnegativity constraints

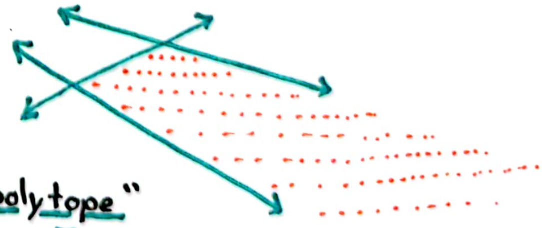
as $x_i \geq 0$ if and only if $e_i^T x \geq 0$

$\uparrow$ i^{th} basis vector $(0,0,\ldots,0,1,0,\ldots,0)$

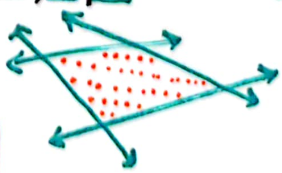
(3) A set given by a single inequality $\{x \mid a^T x \leq b\}$
is called a "halfspace".



(4) A set given by finitely many halfspaces $\{x \mid a_i^T x \leq b_i\}$
is called a "polyhedron".



(5) A bounded polyhedron is called a "polytope"



[The Geometry and Structure of polyhedrons
is explored in detail in "Intro to Convexity".]

We say an LP is <u>infeasible</u> if no $x \in \mathbb{R}^n$ has $Ax \leq b$,

denoted by $\begin{cases} \min & c^T x \\ \text{s.t.} & Ax \leq b \end{cases} = +\infty$ or $\begin{cases} \max & c^T x \\ \text{s.t.} & Ax \leq b \end{cases} = -\infty$.

We say an LP is <u>unbounded</u> if there exists a sequence $x^{(i)} \in \mathbb{R}^n$, $Ax^{(i)} \leq b$

such that $\lim c^T x^{(i)} = -\infty$ (when minimizing)

or $\lim c^T x^{(i)} = +\infty$ (when maximizing)

denoted by $\begin{cases} \min & c^T x \\ \text{s.t.} & Ax \leq b \end{cases} = -\infty$ or $\begin{cases} \max & c^T x \\ \text{s.t.} & Ax \leq b \end{cases} = +\infty$.

We say $x^* \in \mathbb{R}^n$ is a <u>minimizer</u> (<u>maximizer</u>) if $Ax^* \leq b$

and $c^T x^* \underset{(\geq)}{\leq} c^T x$ for all x with $Ax \leq b$.

<u>Common terminology:</u> any point $x \in \mathbb{R}^n$ is a "solution",

any x with $Ax \leq b$ is a "feasible solution",

any minimizing/maximizing x is an "optimal solution".

<u>Theorem</u> Every linear program is either infeasible, unbounded, or has an optimal solution.

Note this does not hold for nonlinear optimization.

Consider minimizing e^x over all of $\mathbb{R}$.

This is feasible, bounded below, and has no minimizer.

We say an LP is in <u>standard form</u> if

$$\begin{cases} \min & c^T x & \text{(always minimizing)} \\ \text{s.t.} & Ax = b & \text{(all equality constraints involving } A\text{)} \\ & x \geq 0 & \text{(all entrywise nonnegative variables).} \end{cases}$$

For example, we can rewrite our grading linear program as

$$\begin{cases} \min & -\left(\frac{(c_H - c_P)}{100} H + \frac{(c_M - c_P)}{100} M + \frac{(c_F - c_P)}{100} F + c_P\right) \\ \text{s.t.} & H + M + F + s_1 = 100 \\ & H - s_2 = 15 \\ & M - s_3 = 15 \\ & -M + F - s_4 = 0 \\ & M + F - s_5 = 50 \\ & M + F + s_6 = 80 \\ & H + M + F - s_7 = 90. \\ & H, M, F, s_1, \ldots, s_7 \geq 0 \end{cases}$$

<u>Theorem (You will prove in HW1)</u> Every LP can be equivalently rewritten into standard form.

[Note equivalent here means if you had an optimal solution to either linear program, you can immediately produce an optimal solution to the other.]

via some simple change of variables

Example Linear Programming Application

Two Constraint Knapsack

Suppose you have goods $i = 1, \dots, m$, (think you are packing apples, bananas, celery, etc.)
each with three attributes:

$$\begin{cases} \text{weight per unit} & w_i \\ \text{volume per unit} & v_i \\ \text{payoff per unit} & p_i . \end{cases}$$

Given weight and volume upper bounds W, V, (think the strength and size of your knapsack)
maximizing your payoff is a linear program:

$$\begin{cases} \max \ p^T x & \left(= \sum_{i=1}^{m} p_i x_i\right) \\ \text{s.t.} \ w^T x \le W & (\Sigma w_i x_i \text{ is at most } W) \\ \phantom{\text{s.t.}} \ v^T x \le V & (\Sigma v_i x_i \text{ is at most } V) \\ \phantom{\text{s.t.}} \ \ \ x \ge 0 & (\text{no negative quantities}) \end{cases}$$

where x_i, our decision variable, is the # units of good i packed.

Notes ▸ Put in standard form, this is

$$\begin{cases} \min \ -p^T x \\ \text{s.t.} \ w^T x + s_1 = W \\ \phantom{\text{s.t.}} \ v^T x + s_2 = V \\ \phantom{\text{s.t.}} \ x, s_1, s_2 \ge 0. \end{cases}$$

▸ Each extreme point has at most two goods packed. Why?

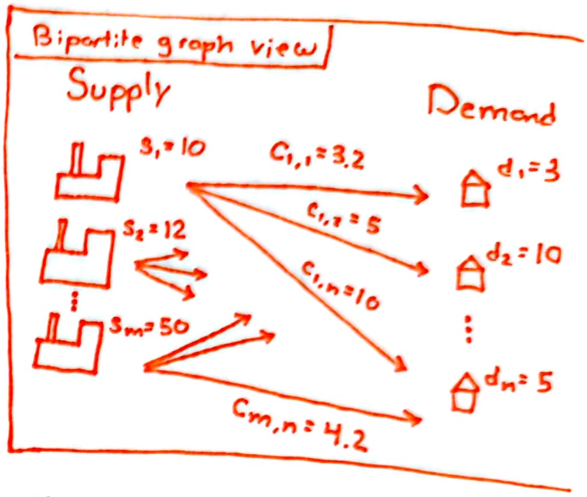
▸ The Dual of this standard form LP is

$$\begin{cases} \max \ W y_1 + V y_2 \\ \text{s.t.} \ w_i y_1 + v_i y_2 \le -p_i \quad \forall i = 1 \dots m \\ \phantom{\text{s.t.}} \ y_1, y_2 \le 0. \end{cases}$$

Why?

Example Linear Programming Application

Transportation Problem



Suppose you are shipping goods from supply centers $i = 1 \dots m$ to demand centers $j = 1 \dots n$.

Each supply center i has s_i goods.
Each demand center j needs d_j goods.
Shipping one unit from i to j costs c_{ij} dollars.

Minimizing your total costs while meeting all demand (assume $\sum_i s_i = \sum_j d_j$) is the following linear program:

$$\begin{cases} \min & \sum_{ij} c_{ij} x_{ij} & & \\ \text{s.t.} & \sum_j x_{ij} = s_i & \forall i = 1 \dots m & \text{(center } i \text{ sends its stock)} \\ & \sum_i x_{ij} = d_j & \forall j = 1 \dots n & \text{(center } j \text{ recieves its demand)} \\ & x \geq 0 & & \text{(no negative shipping)} \end{cases}$$

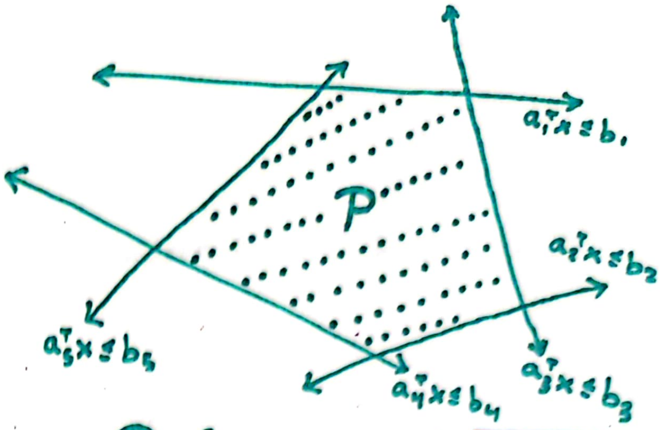
where x_{ij} denotes the amount shipped from i to j.

Notes

- Already in standard form.
- Each extreme point just uses $m+n-1$ shipping links (despite their being nm total options) which form a spanning tree of the bipartite graph above.
  Why?
- The Dual of this standard form LP is $\begin{cases} \max & s^T u + d^T v \\ \text{s.t.} & u_i + v_j \leq c_{ij} \quad \forall i,j. \end{cases}$
  Why?

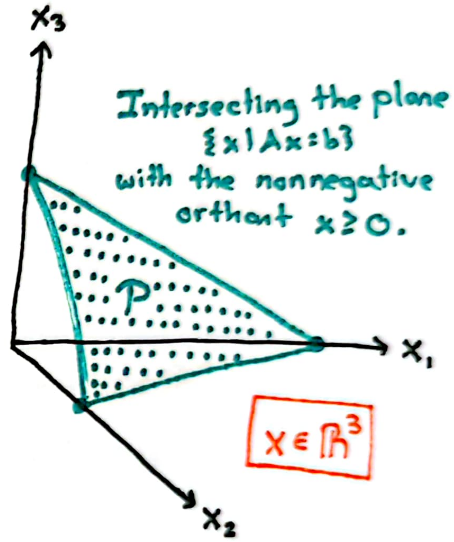
2. Extreme Points

For the sake of visualizations, a general polyhedron looks like...

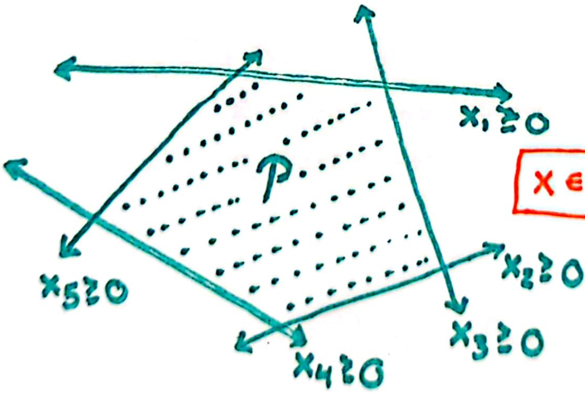


$P = \{x \mid Ax \le b\}$ $\boxed{x \in \mathbb{R}^2}$

A standard form polyhedron is more limited. They all look like...



If we want visualize standard form LPs with dimension >3, we can restrict our view to the subspace $\{x \mid Ax = b\}$...

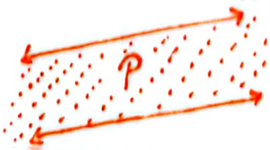


$\boxed{x \in \{x \mid Ax = b\} \subseteq \mathbb{R}^5}$

Note: We can always reformulate/change variables to be in standard form.

This gives us a very nice property: Why?

"$P = \{x \mid Ax = b, x \ge 0\}$ contains no lines $\{x \mid x_0 + \lambda v = x \text{ for some } \lambda\}$ for any x_0, v."

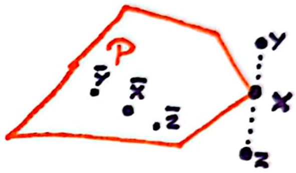
[That is, no shapes like  exist in standard form.]

Three definitions for "corner points"

Let $P = \{x \mid Ax \leq b\}$ be a generic polyhedron ($x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$).

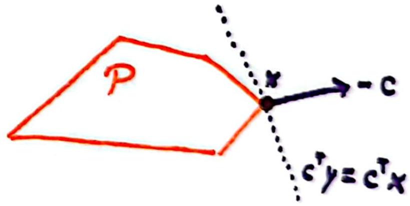
Definition 1 We say $x \in P$ is an extreme point if no $y, z \in P \setminus \{x\}$ exist with $x = \lambda y + (1-\lambda) z$, $\lambda \in [0,1]$.

This has a geometric flavor: $\bar{x}$ is not extreme as shown by $\bar{y}$ and $\bar{z}$. x is extreme as any points y, z it is between have at least one infeasible.



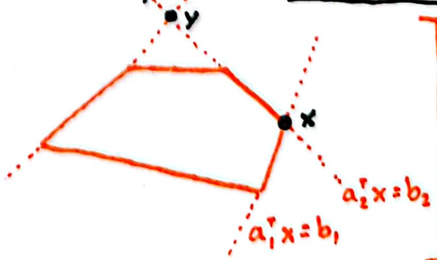
Definition 2 We say $x \in P$ is a vertex if some $c \in \mathbb{R}^n$ has $c^T x < c^T y$ for all $y \in P \setminus \{x\}$.

This has an optimization flavor: x is the unique minimizer in some direction c.



Definition 3 We say $x \in P$ is a Basic Feasible Solution (BFS) if there exist n linearly independent a_i with $a_i^T x = b_i$.
(if we drop the requirement $x \in P$, we say it is a Basic Solution.)

This has an algebraic flavor: x is uniquely determined by a system of n equations. x is a BFS, y is a BS.



Theorem For any polyhedron $P = \{x \mid Ax \le b\}$, a point $x \in \mathbb{R}^n$ is an extreme point iff it is a vertex iff it is a BFS.

Proof. (Vertex $\Rightarrow$ Extreme Point)

Suppose x is a vertex, uniquely minimizing in some direction c:

$$c^T x < c^T y \quad \forall y \in P \setminus \{x\}.$$

Consider any $y, z \in P \setminus \{x\}$, and any $\lambda \in [0,1]$.

Then

$$\begin{array}{rl} & \lambda \cdot (c^T x < c^T y) \\ + & (1-\lambda) \cdot (c^T x < c^T z) \\ \hline & c^T x < \lambda c^T y + (1-\lambda) c^T z \\ & \quad = c^T(\lambda y + (1-\lambda) z). \end{array}$$

Hence $x \ne \lambda y + (1-\lambda) z$.

(Extreme Point $\Rightarrow$ BFS)

Suppose x is not a BFS.

Let $I \subseteq \{1 \dots m\}$ denote the set of "tight" constraints $a_i^T x = b_i$.

$\{a_i\}_{i \in I}$ does not have n linearly independent vectors.

$\Rightarrow$ Some $d \ne 0$ has $a_i^T d = 0$ for all $i \in I$.

$$\Rightarrow \quad \begin{array}{l} a_i^T(x + \varepsilon d) = a_i^T x = b_i \\ a_i^T(x - \varepsilon d) = a_i^T x = b_i \end{array} \quad \forall i \in I$$

Moreover, $a_j^T(x \pm \varepsilon d) = a_j^T x \pm \varepsilon a_j^T d < b_j \quad \forall j \notin I$

if we select ε small enough since $a_j^T x < b_j$.

$\Rightarrow$ x is the average of two feasible points, $x \pm \varepsilon d$.

(BFS $\Rightarrow$ Vertex)

Suppose x is a BFS.

Let $I = \{i \mid a_i^T x = b_i\}$ be the same as above.

Consider minimizing over P in the direction $-c = \sum_{i \in I} a_i$.

All $y \in P$ have $a_i^T y \le b_i$

$$\Rightarrow c^T y = -\sum_{i \in I} a_i^T y \ge -\sum_{i \in I} b_i$$

Equality only holds if $a_i^T y = b_i$ $\forall i \in I$

$$= -\sum_{i \in I} a_i^T x$$

$$= c^T x.$$

$\Rightarrow$ x minimizes $c^T x$ over P.

Since any minimizer as equality above, and $a_i^T y = b_i$ uniquely is solved by x, x is the unique minimizer. $\square$

Corners of Standard Form Polyhedrons

It suffices to consider linear programs over polyhedrons

$$P = \{x \mid Ax = b,\ x \ge 0\}.$$

Without loss of generality. A has independent rows.

A BFS comes from selecting n linearly ind constraints to hold tightly.

- $Ax = b$ gives us m linearly independent constraints
- Some nonnegativity constraints $x_i = 0$ must provide $n-m$ more.

Some notation for standard form BFS

We pick a <u>Basis</u> $B = \{B(1), \dots, B(m)\} \subseteq \{1, \dots, n\}$
and denote its complement by $\bar{B}$.

Let $x_B = (x_{B(1)}, \dots, x_{B(m)})$, $c_B = (c_{B(1)}, \dots, c_{B(m)})$

$x_{\bar{B}} = (x_{\bar{B}(1)}, \dots, x_{\bar{B}(n-m)})$, $c_{\bar{B}} = (c_{\bar{B}(1)}, \dots, c_{\bar{B}(n-m)})$

$$A_B = \begin{bmatrix} | & & | \\ A_{B(1)} & \cdots & A_{B(m)} \\ | & & | \end{bmatrix}$$

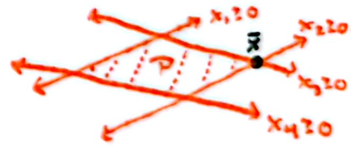
$$A_{\bar{B}} = \begin{bmatrix} | & & | \\ A_{\bar{B}(1)} & \cdots & A_{\bar{B}(n-m)} \\ | & & | \end{bmatrix}.$$

Then the BFS corresponding to B is the unique solution to

$$\begin{cases} Ax = b \\ x_{\bar{B}} = 0 \end{cases} \iff \begin{cases} A_B x_B + A_{\bar{B}} x_{\bar{B}} = b \\ x_{\bar{B}} = 0 \end{cases}$$

$$\iff \begin{cases} A_B x_B = b \\ x_{\bar{B}} = 0. \end{cases}$$

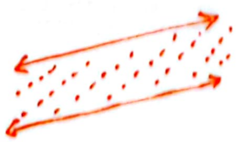
Suppose $B = \{1, 4\}$
$\bar{B} = \{2, 3\}$.
This looks like...



$\{x \mid Ax = b\}$

This has unique solution $x_B = A_B^{-1} b$, $x_{\bar{B}} = 0$ when A_B is invertible and no unique solution otherwise.

<u>Lemma (in HW1)</u> Every nonempty standard form polyhedron has a BFS.

[Recall  is a polyhedron with no extreme points.]

Natural Question

We have seen each standard form BFS corresponds to picking a Basis B and solving $\begin{cases} A_B x_B = b \\ x_{\bar{B}} = 0. \end{cases}$

picking the $x_i \geq 0$ to be tight for $\bar{B}$.

Is this choice of a basis unique?

No!

Two basis B and B' can have the same BFS x solve

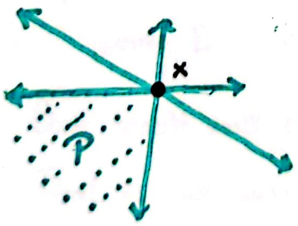
$$\begin{cases} A_B x_B = b \\ x_{\bar{B}} = 0 \end{cases} \quad \text{and} \quad \begin{cases} A_{B'} x_{B'} = b \\ x_{\bar{B}'} = 0. \end{cases}$$

In this case, all $i \in \bar{B} \cup \bar{B}'$ must have $x_i = 0$.

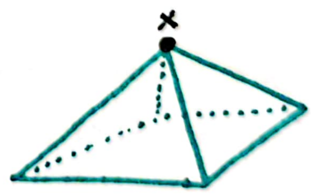
In particular, some $i \in B \setminus B'$ has $x_i = 0$.

(we are getting zeros in the basis that we did not force to be zero.)

For example,



x is determined by any two of the three tight constraints.



x is determined by any three of the four tight constraints.

We say a basis B gives a <u>Degenerate BFS</u> if some $i \in B$ has $x_i = 0$.

We say it is <u>Nondegenerate</u> otherwise (i.e., when $x_B = A_B^{-1} b > 0$).

3. Optimality and Strong Duality

Consider a standard form LP $\begin{cases} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0. \end{cases}$ (LP)

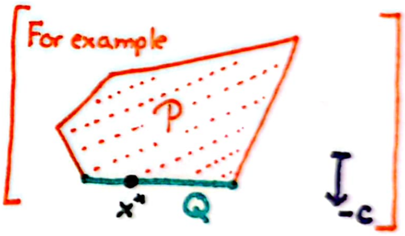
Recall x^* is a minimizer if $\begin{cases} Ax^* = b,\ x^* \geq 0, \\ \text{and } c^T x^* \leq c^T y \quad \forall y \in P = \{x \mid Ax = b,\ x \geq 0\}. \end{cases}$

First we show BFS minimizers typically exist.

Theorem If (LP) has a minimizer, then some BFS is a minimizer.

Proof. Let x^* be a minimizer of $c^T x$ over $P = \{x \mid Ax = b, x \geq 0\}$.

Define the set of minimizers as $Q = \{x \mid Ax = b,\ c^T x = c^T x^*,\ x \geq 0\}$.



Note Q is also in standard form.

By our previous lemma, Q must have some BFS $\bar{x}$.

Our result will then follow if we can show $\bar{x}$ is also a BFS of P.
That is, we want to show no $y, z \in P \setminus \{\bar{x}\}$ have $\bar{x} = \lambda y + (1-\lambda) z$, for any $\lambda \in [0,1]$.

Since $\bar{x}$ is a BFS of Q, no $y, z \in Q \setminus \{\bar{x}\}$ have this.

However if either y or z is in $P \setminus Q$, then $c^T y > c^T \bar{x}$ or $c^T z > c^T \bar{x}$.

$$\Rightarrow \lambda c^T y + (1-\lambda) c^T z > c^T \bar{x}$$
$$\Rightarrow c^T (\lambda y + (1-\lambda) z) > c^T \bar{x}$$
$$\Rightarrow \bar{x} \neq \lambda y + (1-\lambda) z.$$

Thus no $y, z \in P \setminus \{\bar{x}\}$ have weighted average $\bar{x}$. $\square$

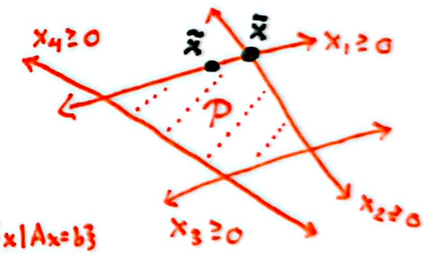
Now we show how to check if a BFS is optimal

Consider any BFS $\bar{x}$ with a corresponding basis B

(that is, $\bar{x}$ uniquely solves $\begin{cases} Ax = b \\ x_{\bar{B}} = 0 \end{cases}$).

For any $j \in \bar{B}$, lets consider slightly relaxing the requirement $x_j = 0$ to $x_j = \varepsilon$, for small $\varepsilon > 0$.

Suppose $\bar{B} = \{1,2\}$, $j = 2$.
Then this looks like...



Let $\tilde{x}$ uniquely solve $\begin{cases} Ax = b \\ x_j = \varepsilon \\ x_{\bar{B} \setminus \{j\}} = 0. \end{cases}$

We can solve this system directly:

$$\begin{cases} Ax = b \\ x_j = \varepsilon \\ x_{\bar{B} \setminus \{j\}} = 0 \end{cases} \iff \begin{cases} A_B x_B + A_j x_j + A_{\bar{B} \setminus j} x_{\bar{B} \setminus j} = b \\ x_j = \varepsilon \\ x_{\bar{B} \setminus j} = 0 \end{cases}$$

$$\iff \begin{cases} A_B x_B = b - A_j \varepsilon \\ x_j = \varepsilon \\ x_{\bar{B} \setminus j} = 0 \end{cases}$$

$$\iff \begin{cases} x_B = A_B^{-1}(b - A_j \varepsilon) \\ x_j = \varepsilon \\ x_{\bar{B} \setminus j} = 0. \end{cases}$$

As we move ε amount, going from $\bar{x}$ to $\tilde{x}$, our objective value changes linearly...

$$c^T\tilde{x} = \begin{bmatrix} c_B \\ c_j \\ c_{\bar{B}\setminus j} \end{bmatrix}^T \begin{bmatrix} A_B^{-1}(b - A_j\varepsilon) \\ \varepsilon \\ 0 \end{bmatrix} = c_B^T A_B^{-1}(b - A_j\varepsilon) + c_j\varepsilon$$
$$= c_B^T A_B^{-1} b + (c_j - c_B^T A_B^{-1} A_j)\varepsilon$$
$$= c^T\bar{x} + (c_j - c_B^T A_B^{-1} A_j)\varepsilon.$$

Using that $x_B = A_B^{-1}b$ and $c^T\bar{x} = c_B^T\bar{x}_B$.

If $c_j - c_B^T A_B^{-1} A_j < 0$, we are improving on $\bar{x}$.

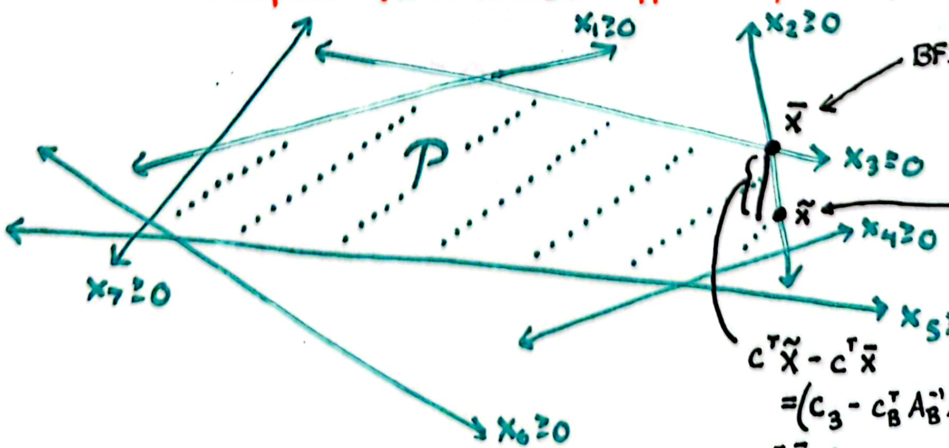
This rate of objective change is important enough to have its own name. The <u>Reduced Cost of j in basis B</u> is $\boxed{\bar{c}_j = c_j - c_B^T A_B^{-1} A_j}$

Note every $i \in B$ has $\bar{c}_i = c_i - c_B^T A_B^{-1} A_i = c_i - c_B^T e_i = c_i - c_i = 0$.

We let $\bar{c}^T$ denote the vector of all reduced costs.
$\bar{c}^T = c^T - c_B^T A_B^{-1} A$

A Picture Recapping the process of relaxing $x_j = 0$ to generate $\tilde{x}$

Suppose $x \in \mathbb{R}^7$, $A \in \mathbb{R}^{5\times 7}$, $b \in \mathbb{R}^5$, and the paper shows the 2D affine subspace $\{x \mid Ax = b\}$. Suppose c points up $\uparrow$, so we minimize down.



BFS with $B = \{1,4,5,6,7\}$
$\bar{B} = \{2,3\}$.

Relaxing $x_3 = 0$:
$\tilde{x} = \bar{x} + \begin{bmatrix} -A_B^{-1}A_3\varepsilon \\ \varepsilon \\ 0 \end{bmatrix}$

$c^T\tilde{x} - c^T\bar{x}$
$= (c_3 - c_B^T A_B^{-1} A_3)\varepsilon$
$= \bar{c}_3\varepsilon$
≥ 0 in this case since $\uparrow c$.

Theorem Consider any BFS x^* with a basis B and reduced costs $\bar{c}$.

(i) If $\bar{c} \geq 0$, x^* is a minimizer.

(ii) If x^* is a minimizer and nondegenerate, $\bar{c} \geq 0$.

Proof. Suppose $\bar{c} \geq 0$. Consider any feasible point $x \in P = \{x \mid Ax = b, x \geq 0\}$.

Proof of (i)

Let $d = x - x^*$.

Since $Ax = b$, $Ax^* = b$, $Ad = 0$.

$$\Rightarrow A_B d_B + A_{\bar{B}} d_{\bar{B}} = 0$$

$$\Rightarrow A_B d_B = -\sum_{j \in \bar{B}} A_j d_j$$

$$\Rightarrow d_B = -\sum_{j \in \bar{B}} A_B^{-1} A_j d_j .$$

Then x^* must be a minimizer since

$$c^T(x - x^*) = c^T d = c_B^T d_B + c_{\bar{B}}^T d_{\bar{B}}$$

by the above calculation

$$= -\sum_{j \in \bar{B}} \left(c_B^T A_B^{-1} A_j d_j - c_j d_j \right)$$

by definition

$$= \sum_{j \in \bar{B}} \bar{c}_j d_j$$

using that $d_j = x_j - x_j^* \geq 0$ as $x_j \geq 0$, $x_j^* = 0$ and that $\bar{c}_j \geq 0$ by assumption.

$$\geq 0 .$$

Proof of (ii)

Suppose some $\bar{c}_j < 0$.

For some $\varepsilon > 0$, consider the $\tilde{x}$ uniquely solving $\begin{cases} Ax = b \\ x_j = \varepsilon \\ x_{\bar{B}\setminus j} = 0 \end{cases}$

which we previously calculated as

$$\begin{bmatrix} \tilde{x}_B \\ \tilde{x}_j \\ \tilde{x}_{\bar{B}\setminus\{j\}} \end{bmatrix} = x^* + \begin{bmatrix} -A_B^{-1}A_j \\ 1 \\ 0 \end{bmatrix} \varepsilon .$$

Since x^* is assumed to be nondegenerate, $x_B^* > 0$ (strictly!).

$\Rightarrow$ For small enough $\varepsilon > 0$, $\tilde{x}_B > 0$.

$\Rightarrow$ Since $\tilde{x}_j = \varepsilon > 0$ and $\tilde{x}_{\bar{B}\setminus\{j\}} = 0$, $\tilde{x} \geq 0$.

$\Rightarrow$ $\tilde{x}$ is feasible for small enough $\varepsilon > 0$.

However, $c^T\tilde{x} = c^Tx^* + \bar{c}_j\,\varepsilon < c^Tx^*$ for all $\varepsilon > 0$.

$\Rightarrow$ x^* is not optimal. $\square$

Note the requirement of nondegeneracy in (ii) is needed.

Designing a simple example with this property failing at a degenerate BFS is a good exercise.

Back in lecture 1, we understood optimality by picking multipliers (magically) for each constraint, giving a dual optimization problem. These reduced costs are actually getting at the same quantities.

Let's retrace our multiplier approach from the grading LP on a generic standard form linear program.

Constructing the dual of a standard form LP

Consider the primal problem $\begin{cases} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0. \end{cases}$ (recall $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$).

Pick any multipliers $y \in \mathbb{R}^m$ for each equality constraint.

Summing these up weighted gives $\sum_{i=1}^{m} y_i (a_i^T x = b_i)$

$$\Rightarrow y^T A x = b^T y. \qquad (1)$$

If we pick multipliers with $y^T A \leq c^T$ elementwise, then $x \geq 0$ ensures $y^T A x \leq c^T x$. (2)

(1)+(2) implies any multipliers $A^T y \leq c$ bound the primal minimization as having $c^T x \geq b^T y$ $\forall$ feasible x.

Computing the largest lower bound is the dual problem $\begin{cases} \max & b^T y \\ \text{s.t.} & A^T y \leq c. \end{cases}$

Theorem (Weak Duality) For any (A,b,c)

$$\left\{\begin{array}{ll}\min & c^T x \\ \text{s.t.} & Ax=b \\ & x\geq 0\end{array}\right. \;\geq\; \left\{\begin{array}{ll}\max & b^T y \\ \text{s.t.} & A^T y\leq c.\end{array}\right.$$

Proof. Immediate from our proceeding construction. $\square$

Theorem (Strong Duality) For any (A,b,c), if at least one of the primal or the dual LP is feasible, then

$$\left\{\begin{array}{ll}\min & c^T x \\ \text{s.t.} & Ax=b \\ & x\geq 0\end{array}\right. \;=\; \left\{\begin{array}{ll}\max & b^T y \\ \text{s.t.} & A^T y\leq c.\end{array}\right.$$

Proof. Coming up in the next lecture or two by analyzing the "Simplex Method". $\square$
It will suffice to find a BFS with $\bar{c}\geq 0$.

Connect Reduced Costs of a Basis to Dual Solutions

Recall the reduced costs of a basis B are $\bar{c}=c-c_B^T A_B^{-1}A$,
and optimality holds if $\boxed{\bar{c}^T = c^T - c_B^T A_B^{-1} A \geq 0.}$

Dual feasibility of some $y\in\mathbb{R}^n$ is $\boxed{c-A^T y\geq 0}$

Picking $y=A_B^{-T}c_B$ makes these two equivalent.

Hence a basis B gives primal solutions $\left\{\begin{array}{l}x_B=A_B^{-1}b\\ x_{\bar{B}}=0\end{array}\right.$ and dual solution $y=A_B^{-T}c_B$.

These have equal objective: $c^T x = c_B^T A_B^{-1} b + c_{\bar{B}}^T 0 = b^T A_B^{-T} c_B = b^T y$.

$\Rightarrow$ Strong duality holds if we can find B with x,y both feasible.

$\boxed{\text{An aside}}$ Duality can be defined more generally than the previously calculated standard form.

Consider a linear program
$$\begin{cases} \min & c^T x + d^T u \\ \text{s.t.} & Ax + Bu = b \\ & Cx + Du \geq e \\ & x \geq 0, \; u \text{ free.} \end{cases}$$

Define multipliers y for the equality constraints and $v \geq 0$ for the inequality constraints.

Summing up the weighted constraints, we find

$$y^T A x + y^T B u + v^T C x + v^T D u \geq b^T y + e^T v$$
$$\| $$
$$(A^T y + C^T v)^T x + (B^T y + D^T v)^T u$$

If we select multipliers with $A^T y + C^T v \leq c$, then $(A^T y + C^T v)^T x \leq c^T x$.
If we select multipliers with $B^T y + D^T v = d$, then $(B^T y + D^T v)^T u = d^T u$.
So any y, v satisfying these gives a lowerbound $c^T x + d^T u \geq b^T y + e^T v$.

Thus the largest lower bound is given by the dual linear program...

$$\begin{cases} \max & b^T y + e^T v \\ \text{s.t.} & A^T y + C^T v \leq c \\ & B^T y + D^T v = d \\ & y \text{ free}, \; v \geq 0. \end{cases}$$

4. The Simplex Method

Before proving strong duality, we introduce an iterative algorithm for solving LPs. Showing the correctness of this method will incidentally prove strong duality.

For all of our development here, as a simplification, we assume all BFS are nondegenerate.

First we define "pivoting" a way to move from one BFS to another.

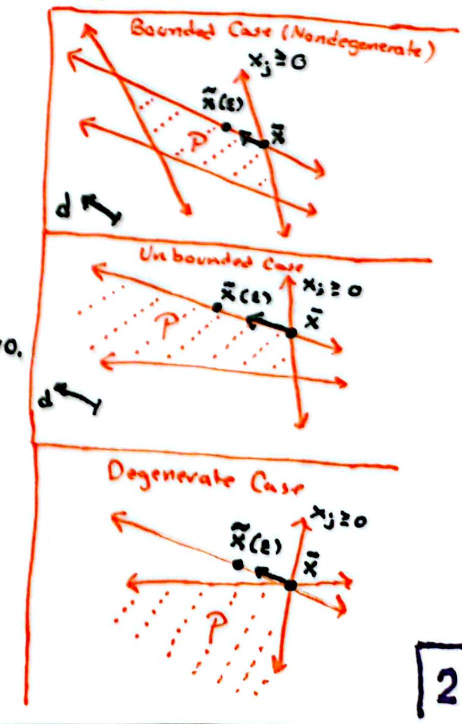
Consider a (nondegenerate) BFS $\bar{x}$ with basis B.

Recall by relaxing $x_j = 0$ to $x_j = \varepsilon$ for $j \in \bar{B}$ defined points

$$\tilde{x}(\varepsilon) \text{ uniquely solving } \begin{cases} Ax = b \\ x_j = \varepsilon \\ x_{\bar{B}\setminus j} = 0 \end{cases} \iff \begin{cases} x_B = \bar{x} - A_B^{-1} A_j \varepsilon \\ x_j = \varepsilon \\ x_{\bar{B}\setminus j} = 0. \end{cases}$$

Note $\tilde{x}(\varepsilon) = \bar{x} + d\varepsilon$ for $d = \begin{bmatrix} -A_B^{-1} A_j \\ 1 \\ 0 \end{bmatrix}$.

On the right we sketch three broadly interesting cases for what this might look like.



Unbounded Case

If $d \geq 0$, then $\tilde{x}(\varepsilon) \geq 0$ $\forall \varepsilon > 0$. So we have an unbounded ray of feasible solutions.

If $\bar{c}_j < 0$, these have $c^T \tilde{x}(\varepsilon) \to -\infty$ as $\varepsilon \to \infty$.

So the LP is unbounded.

Bounded Cases (Nondegenerate and Degenerate)

Suppose $d \not\geq 0$. So at least one $i \in B$ has $d_i < 0$.

$\Rightarrow$ $\tilde{x}(\varepsilon)$ violates $\tilde{x}_i(\varepsilon) \geq 0$ for each such i

if and only if $\tilde{x}_i + \varepsilon d_i < 0 \left(\Leftrightarrow \varepsilon > \frac{\tilde{x}_i}{-d_i}\right)$.

Let $\begin{cases} \varepsilon^* = \min\left\{\frac{\tilde{x}_i}{-d_i} \,\middle|\, d_i < 0\right\} \\ i^* \in \operatorname{argmin}\left\{\frac{\tilde{x}_i}{-d_i} \,\middle|\, d_i < 0\right\} \end{cases}$ denote the maximum ε with $\tilde{x}(\varepsilon)$ feasible and the i limiting this.

[Note if $\tilde{x}$ is degenerate, $\varepsilon^* = 0$ may occur and so no movement occurs.]

Lemma $B' = B \cup \{j\} \setminus \{i^*\}$ is a basis for $\tilde{x}(\varepsilon^*)$.

Proof. Clearly $\tilde{x}(\varepsilon^*)$ solves $\begin{cases} Ax = b \\ x_{\bar{B}'} = 0. \end{cases}$

We need to show it uniquely solves this system.

That is, since $\begin{cases} Ax = b \\ x_{\bar{B}'} = 0 \end{cases} \Leftrightarrow \begin{cases} A_{B'} x_{B'} = b \\ x_{\bar{B}'} = 0, \end{cases}$

we need to show $A_{B'}$ is invertible.

Note $A_{B'} = \left[A_{B(1)} \cdots \underset{\uparrow k^{\text{th}} \text{ column}}{A_j} \cdots A_{B(m)}\right] =$ "A_B with A_{i^*} replaced by A_j"

$= A_B + (A_j - A_{i^*})e_k^T$ $\leftarrow (0, \dots, \underset{\uparrow k^{\text{th}} \text{ element}}{1}, \dots 0)$.

Sherman Morrison says $A_{B'}$ is invertible

iff A_B is invertible and $1 + e_k^T A_B^{-1}(A_j - A_{i^*}) \neq 0$.

This holds since $1 + e_k^T A_B^{-1}(A_j - A_{i^*}) = 1 + e_k^T(-d_B - e_k) = 1 - d_i - 1 = -d_i > 0$. $\square$

Iteratively pivoting to BFS with lower objective values gives a famous algorithm.

The Simplex Method

Given a basis B_0 with primal solution $x(B_0)$ solving $\begin{cases} Ax=b \\ x_{\bar{B}_0}=0 \end{cases}$ primal feasible,

Iterate for $k=0,1,2,\dots$

Compute the dual solution $y(B_k) = A_{B_k}^{-T} c_{B_k}$.

If $y(B_k)$ is dual feasible (i.e. $\bar{c} \geq 0$),

STOP and return primal, dual optimal $x(B_k)$, $y(B_k)$.

Else

Pick any j with $\bar{c}_j < 0$, and compute $d = \begin{bmatrix} -A_{B_k}^{-1} A_j \\ 1 \\ 0 \end{bmatrix}$.

If $d \geq 0$,

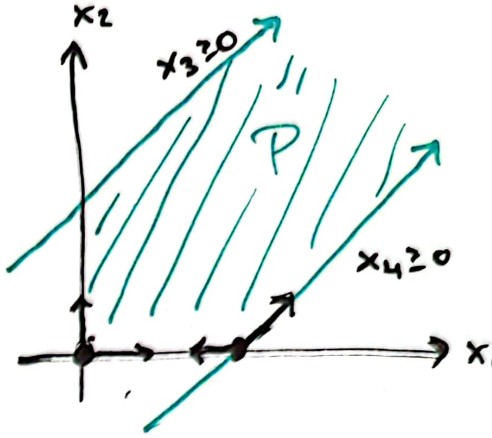
STOP and return LP is unbounded.

Else

$B_{k+1} = B_k \cup \{j\} \setminus \{i\}$ for any i picked from $\operatorname{argmin}\left\{ \frac{x_i(B_k)}{-d_i} \,\middle|\, d_i < 0 \right\}$.

Example Simplex Steps

$$\begin{cases} \min & -x_1 \\ \text{s.t.} & -x_1 + x_2 + x_3 = 1 \\ & -x_1 + x_2 - x_4 = -1 \\ & x \geq 0 \end{cases}$$



The origin is a BFS with basis $B = \{3,4\}$.

$$x = (0,0,1,1)$$

This has reduced cost $\bar{c} = c - A^T A_B^{-T} c_B$

$$= \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} -1 & -1 \\ 1 & 1 \\ 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}^{-T} \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Pick $j = 1$, "pivot" moving along $(0,0,1,1) + \varepsilon \underbrace{(1,0,1,-1)}_{=d}$.

$\varepsilon^* = 1$ attained by $i = 4$. $\Rightarrow B' = \{1,3\}$.

$$x = (1,0,2,0).$$

B' has reduced cost $\bar{c} = c - A^T A_{B'}^{-T} c_{B'}$

$$= \begin{bmatrix} 0 \\ -1 \\ 0 \\ 1 \end{bmatrix}$$

Pick $j = 2$

So moving along $(1,0,2,0) + \varepsilon \underbrace{(1,1,0,0)}_{d}$

$d \geq 0$.

$\Rightarrow$ Unbounded.

Proof of (Linear Programming) Strong Duality

Recall our theorem statement from page 26:

Theorem (Strong Duality) For any (A, b, c), if at least one of the primal or dual LPs is feasible, then

$$\left\{\begin{array}{l}\min\ c^T x \\ \text{s.t. } Ax = b \\ \quad\ \ x \geq 0\end{array}\right. = \left\{\begin{array}{l}\max\ b^T y \\ \text{s.t. } A^T y \leq c.\end{array}\right.$$

If either LP is unbounded, the equality is immediate from weak duality's inequality.

So it suffices to show when one LP is feasible, both are feasible with x^*, y^* existing with (1) primal feasibility $\left\{\begin{array}{l}Ax^* = b \\ x^* \geq 0\end{array}\right.$

(2) dual feasibility $\{A^T y^* \leq c$

(3) optimality $\{c^T x^* = b^T y^*.$

In particular, it suffices to show some basis B has associated $x(B)$ primal feasible and $y(B)$ dual feasible as (3) always holds

$$c^T x(B) = c_B^T A_B^{-1} b = b^T A_B^{-T} c_B = b^T y(B).$$

This is exactly what the Simplex Method constructs!!

Thus it suffices to show the Simplex Method terminates.

Hence we will just argue the Simplex Method never revisits a basis as there are at most $\binom{m}{n}$ of them.

Easy Case: Suppose every extreme point x is nondegenerate (that is, $x_B > 0$).

Then every pivot of the simplex method has

$$\varepsilon^* = \min\left\{ \frac{x_i(B_k)}{-d_i} \;\middle|\; d_i < 0 \right\} > 0 \quad \text{(strictly).}$$

$$\begin{aligned}
\Rightarrow c^T x(B_{k+1}) &= c^T(x(B_k) + \varepsilon^* d) \\
&= c^T x(B_k) + \varepsilon^* c^T d \\
&= c^T x(B_k) + \varepsilon^* \bar{c}_j \\
&< c^T x(B_k) \qquad \text{(strictly!).}
\end{aligned}$$

Thus the objective strictly decreases each step.

$\Rightarrow$ We cannot revisit a BFS.

$\Rightarrow$ Simplex must terminate with a primal, dual pair proving strong duality holds. $\square$

Hard Case: If $x(B_k)$ is degenerate, we may have $\varepsilon^* = 0$.

Then the strict decrease above does not hold.

Under generic pivoting rules for selecting j and i, Simplex may cycle forever.

(all the gorey details here are beyond our scope. An example detailing this problem will be emailed out for those interested in more LP theory.)

To handle degeneracy, we need to make more structured choices of j and i than arbitrary selection:

(Lexigraphical Pivoting)

Bland's Rule for Simplex

Pick $j \in \bar{B}$ with $\bar{c}_j < 0$ and the smallest such index j.

Pick $i \in B$ attaining $\min\left\{ \frac{x_i(B_x)}{-d_i} \,\middle|\, d_i < 0 \right\}$ with the smallest such index i.

Under this rule, one can show eventually $\varepsilon^* > 0$ and so progress is made.

(again full details are beyond our scope but will be emailed out.)

Then the easy case argument can be applied to guarantee primal, dual optimal pairs will eventually be found. $\square$

Note: Better than $x(B), y(B)$ just being optimal, we know that they have

"Complementary Slackness": Each $i \in B$ may have $x_i(B) > 0$ but must have $\bar{c}_i = c_i - A_i^T y(B) = 0$.

Each $j \in \bar{B}$ must have $x_j(B) = 0$ but may have $\bar{c}_j = c_j - A_j^T y(B) > 0$.

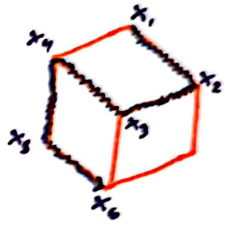
$$\Rightarrow \underbrace{x(B)^T}_{\geq 0}\underbrace{(c - A^T y(B))}_{\geq 0} = 0.$$

6. Comments on Computation

History of Computational Guarantees on Simplex

The Simplex Method may take exponentially many steps (as far as we know) under every pivoting rule we have tried.

Usually show with a "Minty Cube" type example, perturbing a hypercube.



$\approx 2^n$ steps

In 80s + 90s, folks showed simplex on average (over a distribution of all LPs) only needs $O(n)$ pivots.

In 2000s, Smoothed analysis shows simplex only needs a polynomial number of pivots on $Ax = b + \sigma$ with Gaussian noise σ. "Most problems near any problem are polynomial".

To practically apply simplex, we need an initial BFS.

One common solution: First solve an LP seeking feasibility

Given an LP (A, b, c), we have $b \geq 0$ WLOG (negate equality constraints with $b_i < 0$ to have $b_i > 0$).

Consider the auxilliary LP

$$(*) \begin{cases} \min \sum s_i \\ \text{s.t. } Ax + s = b \\ \quad x, s \geq 0. \end{cases}$$

Claims: $(0, b)$ is a BFS of $(*)$.

$(*)$ has optimal value 0 iff the original LP is feasible.

Proof. Left as exercise. $\square$

Some Programming Tools for LPs

CVX is a general structured optimization tool in many languages...

(in python cvxpy
in matlab cvx
in Julia Convex.jl)

For example, in Julia given matrix A and vectors b, c, solving the LP is one line:

```
using Convex
x = Variable()
p=minimize( dot(c,x), A*x = b, x >= 0).
```

Dual multipliers certifying the returned solution are also provided

```
p.dual.
```

Classic Simplex solvers are CPLEX and gurobi, which both offer free academic licenses, but cost for industry.

We will deal with other solvers as needed for more general nonlinear optimization problems.

$\boxed{\text{An aside}}$ On the complexity of representing polyhedra.

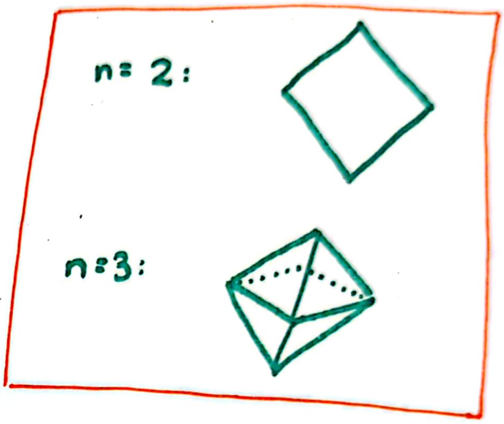
Consider the ℓ_1-ball, $S = \{ x \mid \underbrace{\sum_{i=1}^{n} |x_i|}_{\|x\|_1} \le 1 \}$.

Efficient Representation

This set has $2n$ BFS given by...

$\{ \pm e_1, \dots, \pm e_n \}$

where $e_i = (0, \dots 0, 1, 0, \dots 0)$.
$\uparrow_{i^{th} \text{ position}}$



$\Rightarrow S = \text{convex hull}(\{\pm e_i\})$.

(this description of S is linear in the dimension n)

Inefficient Representation

This set has 2^n faces, each given by an inequality,

$a_i^T x \le 1$ for each $a_i \in \{(\pm 1, \pm 1, \dots, \pm 1)\} = \{\pm 1\}^n$.

$\Rightarrow S = \{ x \mid a_i^T x \le 1 \ \forall a_i \in \{\pm 1\}^n \}$.

(this description of S has exponential size in n)

Takeaway: Some polyhedrons are better represented for computation as $\underbrace{\text{convexhull}(\{p_i\})}$ vs $\{ x \mid Ax \le b \}$.

Solving these LPs are easy: $\min c^T x$ s.t. $x \in \text{convexhull}(\{p_i\})$
$= \min_i c^T p_i$.

An aside on representations, continued

Consider the following NP-Hard problem

$$\begin{cases} \min & x^T A x \\ \text{s.t.} & x \in \{\pm 1\}^n . \end{cases}$$

Combinatorial problems like max-cut, knapsack, traveling salesman, can all be described in this form.

We can rewrite this with a linear objective over a matrix problem ($\langle A, X \rangle$ is the trace inner product $tr(AX)$):

$$\begin{cases} \min & \langle A, xx^T \rangle \\ \text{s.t.} & x \in \{\pm 1\}^n \end{cases}$$

using cyclic property of trace
$tr(x^T A x) = tr(A x x^T) = \langle A, xx^T \rangle$.

$$= \begin{cases} \min & \langle A, X \rangle \\ \text{s.t.} & \underbrace{X \in \text{convexhull}(xx^T \mid x \in \{\pm 1\}^n)}_{\text{polyhedron in matrix space}} \end{cases}$$

(some 3D printed examples are on my website)

This LP is equivalent to an NP-Hard problem.

In this case, we can view this as the polyhedron needing (as far as we know) exponential sized description in terms of either BFS or faces.

In [53]:
```
using LinearAlgebra    #The default package for matrices and such oeprations
using Convex           #A solver interface for convex optimization (including linear pro
gramming)
using SCS              #A solver using ADMM (an algorithm we will discuss later in Nonli
near II)


function gradeStudent(scoreH, scoreM, scoreF, scoreP)
#Given a student's indivudual scores, ranging 0 to 100, in the four course components
#Returns their approximate maximum course grade over all allowable rubircs
#
#Warning the returned solution is only approximately optimal since ADMM only approximate
ly solves

    #Define variable for the LP. The coordinates are the weights (H, M, F, P)
    x = Variable(4)

    #Define the objective for the LP
    c = [scoreH; scoreM; scoreF; scoreP]/100.0
    p = maximize(dot(c, x))

    #Define the problems constraints
    p.constraints += [x[1] + x[2] + x[3] + x[4] == 100;
                      x[1] + x[2] + x[3]        <= 100;
                      x[1]                      >= 15;
                             x[2]               >= 15;
                         -1*x[2] + x[3]         >= 0;
                             x[2] + x[3]        >= 50;
                             x[2] + x[3]        <= 80;
                      x[1] + x[2] + x[3]        >= 90]


    #Run the SCS solver on our newly constructed LP
    solve!(p, SCS.Optimizer; silent_solver = true)

    return p.optval
end
```

Out[53]: gradeStudent (generic function with 1 method)

In [57]:
```
gradeStudent(89, 91, 82, 100)
```

Out[57]: 88.84915267081558