Lec 2.

- There are two approaches to solving an LP problem:

Example. maximize $3x_1+2x_2$

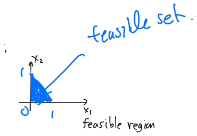
subject to $x_1+x_2\le 1$

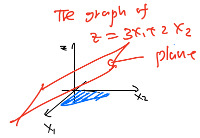
$x_1\ge 0$

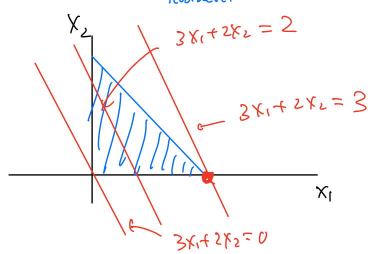
$x_2\ge 0$

<sol>

Geometric method;







geometric

algebraic method: maximize $3x_1+2x_2$

subject to $x_1+x_2\le 1$

$x_1\ge 0$ this inequality condition is inconvenient to handle.

$x_2\ge 0$

Introduce $x_3=1-x_1-x_2$ slack variable.

$x_1+x_2\le 1 \Longleftrightarrow x_1+x_2+x_3=1$

Rewrite the problem: maximize $3x_1+2x_2$

$x_1+x_2+x_3=1$

$x_1\ge 0$

$x_2\ge 0$

How to handle this?

Use $x_1=1-x_2-x_3$

Then $3x_1+2x_2=3(1-x_2-x_3)+2x_2$

$=3-x_2-3x_3$

Then the problem is:

maximize $3-x_2-3x_3$

$x_1+x_2+x_3=1$

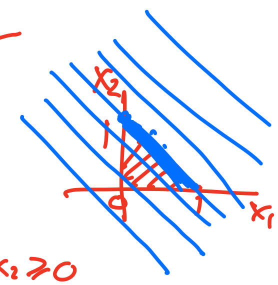
$x_1,x_2,x_3\ge 0.$

So, max at $x_2=0$, $x_3=0$ so at $(x_1,x_2)=(1,0)$ ▨

algebraic

Question:

Note There are LP problems that have multiple optimal solutions: e.g. maximize x_1+x_2 subject to $x_1+x_2\le 1$, $x_1\ge 0$, $x_2\ge 0$.



Q. What are the correct statements?

1) Linear programming (LP) problems are always solvable by algebraic operations such as addition/subtraction, multiplication/division, etc. Yes We will learn 'Simplex method'.
2) Linear programming problems are always solvable by geometric methods, like moving the level sets of the objective function over the feasible region. Yes But only "theoretically".
3) There are linear programming problems that has no solution. Yes. e.g. maximize x subject to $x\ge 1$ and $x\le -1$. not feasible
4) If an LP problem has non-empty feasible region then it has an optimal solution. e.g. "unbounded LP" → maximize x subject to $x\ge 1$ ← feasible but does not have an optimal solution.

So, LP can be solved "easily"
either in geometric & algebraic method.

Really ?

Ex (Dantzig) A planning problem

How to most efficiently assign

N people to N tasks

(Allow a person can do multiple tasks)

Assume: • benefit of assigning person i to task j $= c_{ij} \leftarrow$ some given number

• x_{ij} = the portion of person i's time spent on doing task j

decision variables

$0 \le x_{ij}$ & $\sum_{j=1}^{N} x_{ij} = 1$ for each i

• each job must be done: $\sum_{i=1}^{N} x_{ij} = 1$. for each j

$$\text{maximize} \sum_{j=1}^{N}\sum_{i=1}^{N} c_{ij}\, x_{ij} \quad \leftarrow \text{objective function}$$

subject to

constraints that determine the feasible set

$$\left\{\begin{array}{lll} \sum_{j=1}^{N} x_{1j} = 1 & \sum_{i=1}^{N} x_{i1} = 1 & x_{ij} \ge 0 \\ \sum_{j=1}^{N} x_{2j} = 1 & \sum_{i=1}^{N} x_{i2} = 1 & \forall\ 1 \le i \le N \\ \cdots & \cdots & \quad\ \ 1 \le j \le N \\ \sum_{j=1}^{N} x_{Nj} = 1 & \sum_{i=1}^{N} x_{iN} = 1 & \end{array}\right.$$

$N \times N$ variables

& $N + N + N \times N$ constraints!

For $N = 60$? (e.g. Math dept. of UBC has about 60 regular faculty members)

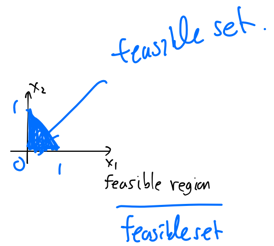
What about $N = 100, 1000, 100{,}000$?

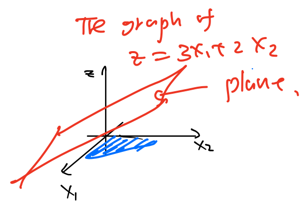
- Key points:
- Practical LP problems are NOT easy to solve due to many variables (high dimensions).
  - Finding a solution can be very costly.
- There are two approaches to solving an LP problem:
  - Geometric
  - Algebraic
- The geometric method is not practical for high dimensional problems.
  - The feasible set is something not visualizable in practice.
- The algebraic method needs to be more systematic to handle MANY variables.
  - We want effective algorithms.

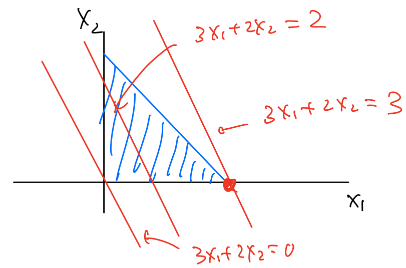
Wisdom for us:
- Use geometry to get intuition for finding effective algebraic algorithms.

- We will learn how to combine geometric intuitions with algebraic algorithms.

Geometric method;

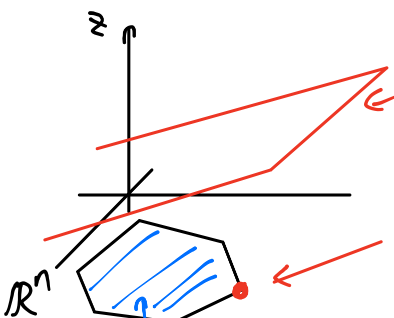






Observations to generalize:

- The feasible set is a polytope.
- The maximum occurs at the vertex (corner) of the polytope.



objective function (which is linear)

$f(x_1, \cdots, x_n) = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n$

The maximum point is a vertex of the polytope.

feasible region by linear constraints is a polytope

high dimensional

We will investigate in this direction: Simplex method. Later.

Self Study Material: "Standard form"

- 2020-340-lec1-self.pdf AND Anstee's lecture note (standard-form-Anstee.pdf) in the file folder Self-Study-Material in the Canvas.
- Matrix description. Vanderbei 6.1

Geometry of linear constraints.

$\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n \qquad \vec{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \in \mathbb{R}^n$ ← vector.

$\vec{a} \cdot \vec{x} = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n$ ← linear function. e.g. $2x_1 + 3x_2$.

Half spaces

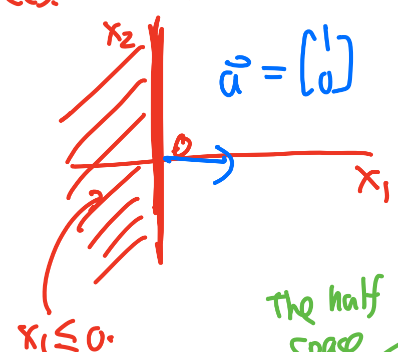
$\{ \vec{x} \mid \vec{a} \cdot \vec{x} \le b \}$. e.g. $2x_1 + 3x_2 \le 1$

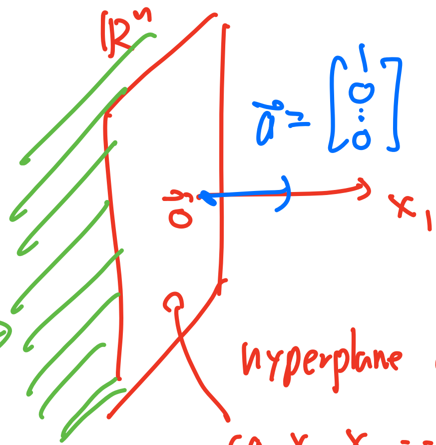
e.g. $\{ \vec{x} \mid x_1 + x_2 + \cdots + x_n \le 1 \}$ $\quad \vec{a} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^n$

e.g. $\{ \vec{x} \mid x_1 \le 0 \}$ $\quad x_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$

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

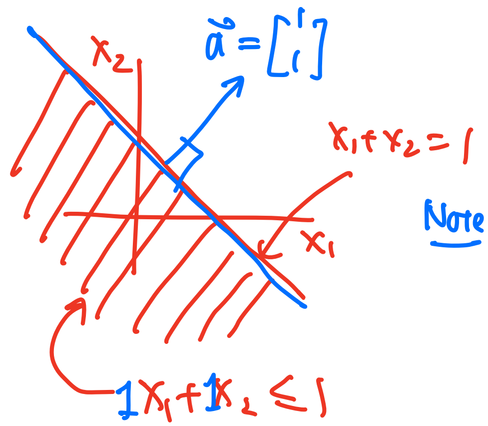
e.g. $n = 2$.



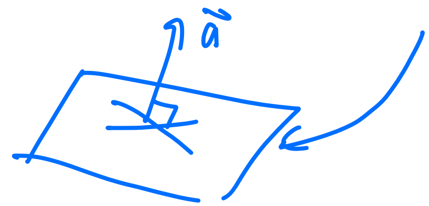


$\{ \vec{x} \mid x_1 \le 0 \}$

$n=2$ $\quad \vec{a} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ orthogonal to the hyperplane $\{ \vec{x} \mid x_1 + x_2 = 1 \}$



Note $\vec{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}$ of $a_1 x_1 + a_2 x_2 + \cdots + a_n x_n$, is perpendicular to the hyperplanes $a_1 x_1 + \cdots + a_n x_n = b$.



$\boxed{\vec{a} \text{ is orthogonal to the hyperplanes } \{ \vec{x} \mid \vec{a} \cdot \vec{x} = d \} = \{ (x_1, x_2, \dots, x_n) \mid a_1 x_1 + \cdots + a_n x_n = d \}}$

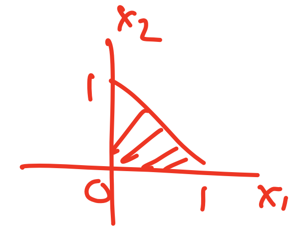
d = constant

Intersections of half spaces

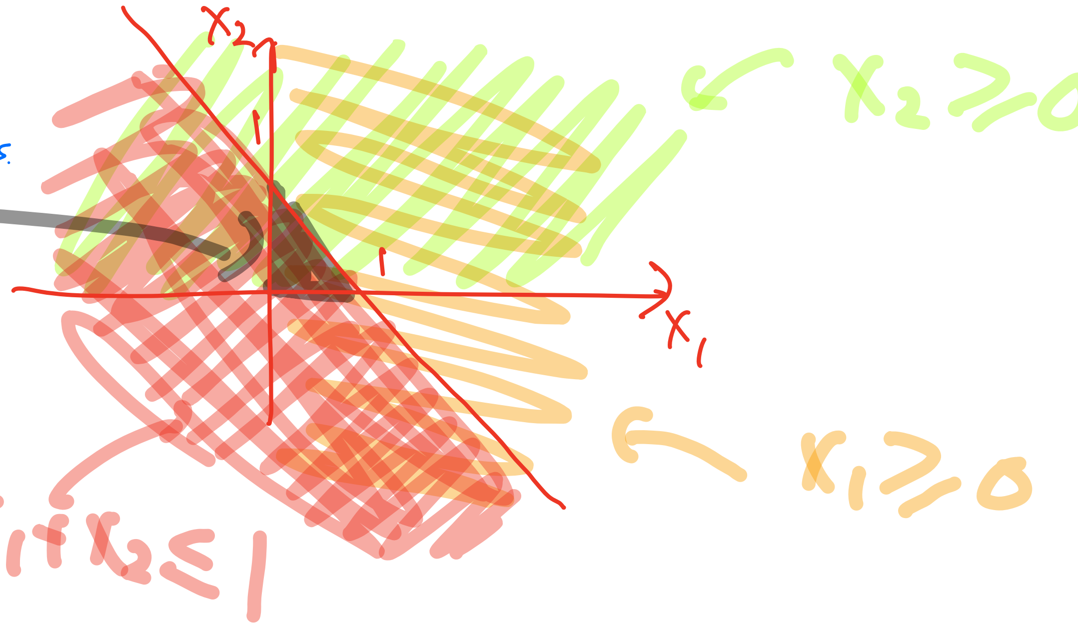
for half spaces $H_i = \{ \vec{x} \in \mathbb{R}^n \mid \vec{a}_i \cdot \vec{x} \le b_i \}, \; i = 1, \cdots, N$

$$H_1 \cap H_2 \cap \cdots \cap H_N$$

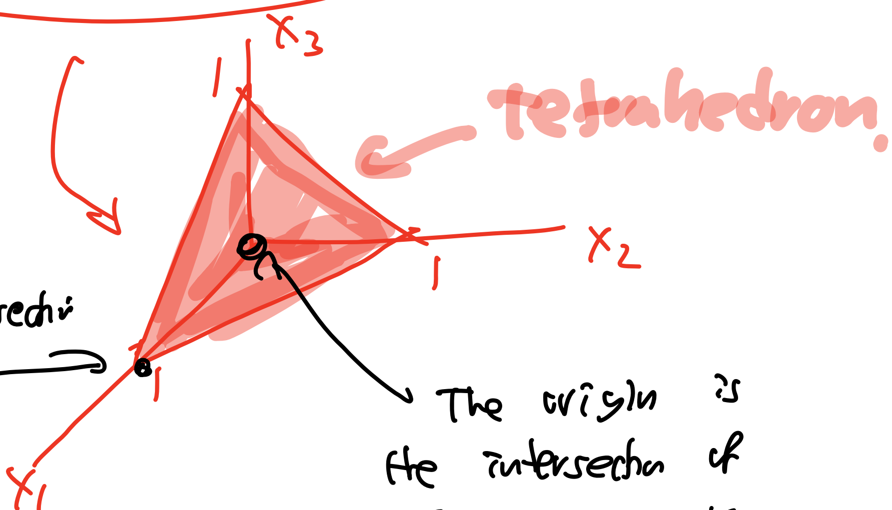
$$\begin{cases} x_1 + x_2 \le 1 \\ x_1 \ge 0 \\ x_2 \ge 0 \end{cases}$$



e.g. $\begin{cases} x_1 + x_2 \le 1 \\ x_1 \ge 0, \\ x_2 \ge 0 \end{cases}$ three linear constraints.



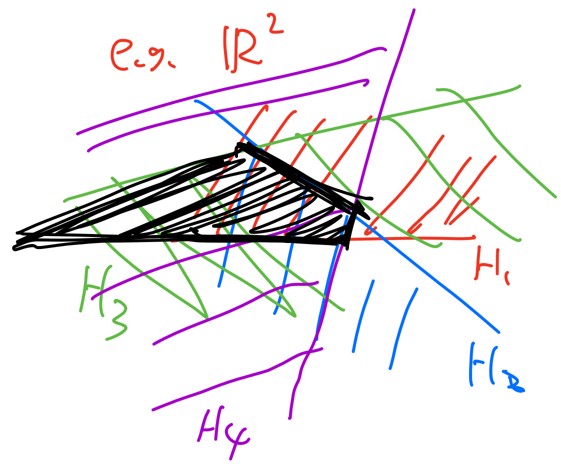
e.g. $\begin{cases} x_1 + x_2 + x_3 \le 1 \\ x_1 \ge 0, \; x_2 \ge 0, \; x_3 \ge 0 \end{cases}$



$C(1,0,0)$ is an intersection between $x_1 + x_2 + x_3 = 1$, $x_2 = 0$, $x_3 = 0$,

The origin is the intersection of $x_1 = 0$, $x_2 = 0$, $x_3 = 0$

e.g. $\mathbb{R}^2$

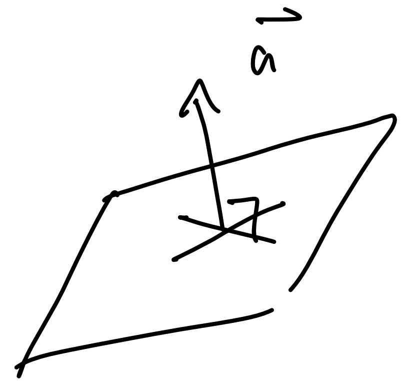


Intersections of hyperplanes

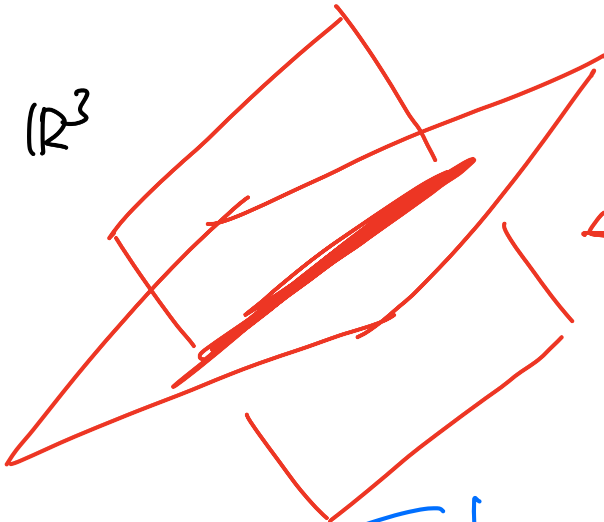
d = constant

hyperplanes $\{ \vec{x} \mid \vec{a} \cdot \vec{x} = d \}$.

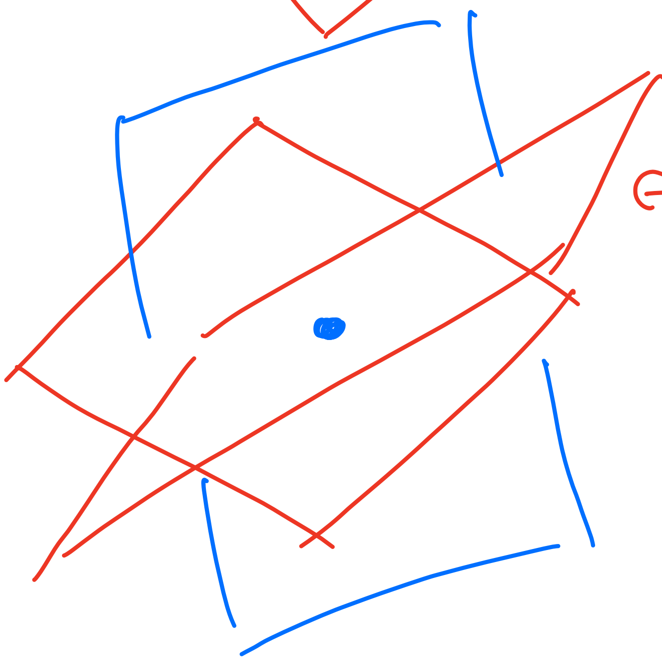
$= \{ (x_1, x_2, \cdots, x_n) \mid a_1 x_1 + \cdots + a_n x_n = d \}$



e.g. $\mathbb{R}^3$



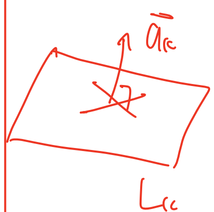
Two planes intersect along a line (only) when their normal vectors are NOT parallel.



Three planes intersect only at a point when there normal vectors, say $\vec{a}_1, \vec{a}_2, \vec{a}_3$ are <u>linearly independent</u>.

General Fact Let $L_1, L_2, \dots, L_m$ be $(n-1)$ dim'l hyperplanes in $\mathbb{R}^n$ with normal vectors $\vec{a}_1, \dots, \vec{a}_m$, respectively

i.e. $L_k = \{\vec{x} \mid \vec{a}_k \cdot \vec{x}_k = b_k\}$, $k = 1, \dots, m$.



Then the intersection
$$L_1 \cap L_2 \cap \dots \cap L_m$$
is an $(n-m)$-dim'l linear space

if the normal vectors $\vec{a}_1, \vec{a}_2, \dots, \vec{a}_m$ of those hyperplanes are linearly independent.

Remark This theorem can be understood from linear algebra by looking at the matrix equation

$$\begin{array}{cc} & \overbrace{\hphantom{a_{11}\ a_{12}\ \cdots\ a_{1n}}}^{n} \\ m\left\{\right. & \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & & & \\ a_{m1} & \cdots & & a_{mn} \end{bmatrix} \end{array} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

which is equivalent to the system of equations

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_n \end{cases}$$

The dimension of the solution set is $n - (\text{rank of the matrix})$.

m of them

For that $m \times n$ matrix A, if the raw vectors $\vec{a}_1, \dots, \vec{a}_m$ are linearly independent then the rank of A is m.

Computational methods.

- Introduction to Jupyter lab/notebook (in UBC Syzygy server), python, and PuLp package.
- The simple examples with PuLp.
- More examples (Blending problem) with Python PuLp package.

UBC Syzygy server.

syzygy.ca is a project of the Pacific Institute for the Mathematical Sciences, Compute Canada and Cybera to bring JupyterLab to researchers, educators and innovators across Canada.

https://ubc.syzygy.ca/ You can log-in with your UBC CWL id.

See the tutorial: https://intro.syzygy.ca

You can instead use this address https://pims.syzygy.ca/ and login with a Google account.
It is a different account, so, your files at UBC account do not show up here, and vice versa.

Jupyter Lab/Notebook.

* Jupyter notebook (https://jupyter.org/) is a user friendly way of writing codes in python programming.

Python. (We use Python 3.)

* For Python language, you can look at a nice introduction by Prof. Patrick Walls at https://www.math.ubc.ca/~pwalls/math-python/

Basics:
How to navigate and run the code.

- Markdown: notes that are not run in the code. Latex symbols work for writing math expressions.
- Code

Lec2.