Linear Programming (LP).

A first example.

Geppetto works makes toy soldiers and toy trains.

	1 soldier	1 train	Available
carpentry (hr)	1	1	80
finishing (hr)	2	1	100
profit ($)	3	2	
limit on demand	40	$+\infty$	

In workshop 1 soldier needs 1 hr of CS, 1 hr of finishing, brings $3 profit.

I can sell at most 40 soldiers.

similar story for trains.

Formulate a linear program

↓

solution: soon!

to maximize profit!

A LP has 3 ingredients:

1) variables → what decision do I make?

2) objective function → what do I optimize?

3) constraints → what are the limitations?

1) x_1 = # of soldiers to make
   x_2 = # of trains to make

2) Max(imize)

$$\underbrace{3x_1}_{\text{profit from soldiers}} + \underbrace{2x_2}_{\text{profit from trains}}$$

3) constraints

$x_1 + x_2 \leq 80$

carpenter need — carpentry available

$2x_1 + x_2 \leq 100$

finishing needed — finishing available

$x_1 \leq 40$

demand for soldiers

$x_1 \geq 0, \quad x_2 \geq 0$

Note: everything is LINEAR.

NO: x_1^2, $x_1 x_2$, $\frac{1}{x_1}$, etc.

Example 2

We want to place commercials on TV.

2 venues

comedy shows
ad costs 60K

football games
ad costs 100K

Target audience: men, women

Each comedy ad is seen by
7 million women
2 million men

Each football ad is seen by
2 million women
12 million men

Goal: ≥ 28 million women &
≥ 24 million men to reach

Set up a LP to accomplish this at minimum cost!

Put data in a table:

	comedy ad	football ad	demand
men (million)	2	12	24
women (million)	7	2	28
cost ($1000)	50	100	

variables (what do we decide?)

x_1 = # of comedy ads placed

x_2 = # of football ads —"—

objective: Min(imize) $50x_1 + 100x_2$

constraints: $2x_1 + 12x_2 \geq 24$ (men constr.)

$7x_1 + 2x_2 \geq 28$ (women constr.)

$x_1 \geq 0, \quad x_2 \geq 0.$

So, what is a linear program?

Linear f

$f: \mathbb{R}^n \to \mathbb{R}$ is a linear fn, if it is of the form

$f(x) = a^T x$, where $a \in \mathbb{R}^n$ is a fixed vector.

Rem: vectors are column vectors in this course.

Ex: $f(x) = x_1 - x_3 + x_4$ is a linear fn.

now: $a = (1, 0, -1, 1, 0 \ldots 0)^T$

$f(x) = 3x_1 + x_2 + x_5$ is a linear fn

now: $a = (3, 1, 0, 0, 1, 0 \ldots 0)^T$

$a = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$ is a column vector

$= (1,$

NOT a linear fn:

$f(x) = 3x_1 - 2x_1 \cdot x_2$

$f(x) = 2x_1 - x_2^2$

A linear constraint

is a constraint of the form

$f(x) \leq \beta$, or

$f(x) \geq \beta$, or

$f(x) = \beta$, where β is a number, f is a linear fn.

Ex: $\left.\begin{array}{l} x_1 + 3x_2 \geq 5 \\ 3x_1 - x_3 \leq 10 \end{array}\right\}$ are linear constr.

$x_1 \geq 0$ is a linear constraint

$2x_1 \geq 3(x_1 + x_2)$ is also

linear constraint after rearranging.

What is NOT a linear constraint:

$x_1 + x_2 < 5$

strict inequalities are NOT permitted!

Why? such problems may not have an optimal solution!

Ex: max x_1

s.t. $x_1 < 1$

↓

subject to

(1) $x_1 \geq 3(x_1 + x_2)$ rewritten as a linear constraint:

$x_1 \geq 3x_2$

(2) $0 \geq 2x_1 + 3x_2$

(1) and (2) are equivalent.

A linear program (LP)

is a problem of maximizing or minimizing a linar fn subject to linar constraints.

Ex: 1) Toy soldiers & trains
2) Ad placement

Ex:

$$\text{Min } x_1 - 2x_2 + x_4$$

s.t.

$$x_1 - x_3 \leq 3$$

$$x_2 + x_4 \geq 2$$

$$x_1 + x_2 = 4$$

$$x_1, x_2, x_3 \geq 0$$

x_4 is free

i.e. not restricted to be ≥ 0.

In an LP we can NOT have:

strict inequalities: $x_1 + x_2 < 2$

nonlinear functions: $\frac{1}{x_1}$, $x_1 \cdot x_2$, x_1^2, ...

"for" loop,

"while" loop, etc.

linear program is not like a C or Java program

History:

Start of LP: WW2 transportation

Importance:

1) we can model many problems as LPs

2) we can solve LPs with 10s of millions of variables and constraints.

We will use software to handle at most 500 variables and constraints together.

if nonlinear functions are involved $\Rightarrow$ nonlinear program.

Trivia: There are nonlinear programs with 4 (four) variables that are unsolvable by even the best computer!

Diet problem and similar

History: LP was used by US military to design meals. Similar problems arise a lot.

Ex: Given 3 types of "food":

Mr Dew, Latte, Cake

	Mr Dew	Latte	Cake
caffeine (mg)	110	200	60
sugar (g)	25	15	30
price ($)	1.5	2.0	1.20

I need $\geq$ 1000 mg of caffeine
$\geq$ 100 g of sugar on a hard day.

Variables:

x_1 = # of bottles of Mt Dew

x_2 = # of bottles

x_3 = # of bottles of Diet Coke

Objective:

$$\min \ 1.5x_1 + 2x_2 + 1.2x_3$$

Constraints:

$$110x_1 + 200x_2 + 60x_3 \geq 1000$$

$$25x_1 + 15x_2 + 30x_3 \geq 100$$

$$x_i \geq 0 \quad \text{for all } i$$

Solve: $x_1 = 0, x_2 = 4.7, x_3 = 0.98$

LPs will often come in matrix form.

Reminder: matrix-vector multiplication

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & & & \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{pmatrix}$$

m rows, n columns.

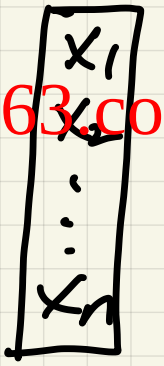
A_{ij} are numbers.

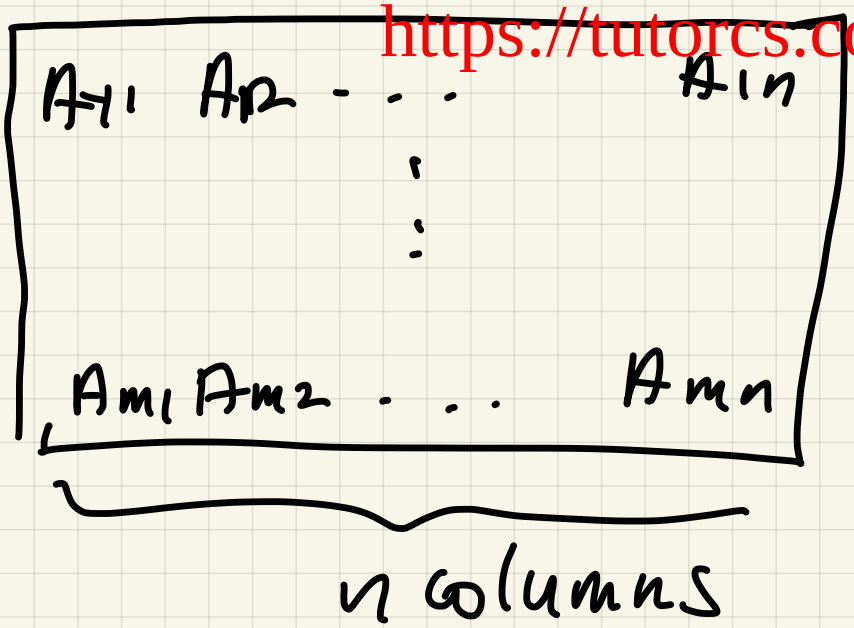
x is column vector of n components. $x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$

Then Ax is a column m-vector

$$Ax = \begin{pmatrix} A_{11}x_1 + A_{12}x_2 + \cdots + A_{1n}x_n \\ A_{21}x_1 + A_{22}x_2 + \cdots + A_{2n}x_n \\ \vdots \\ A_{m1}x_1 + A_{m2}x_2 + \cdots + A_{mn}x_n \end{pmatrix}$$

Visualization:





Ex:

A:

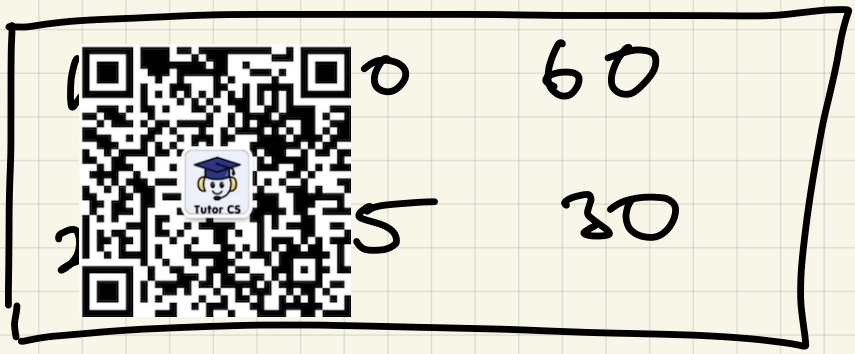
$$\begin{bmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{bmatrix}$$

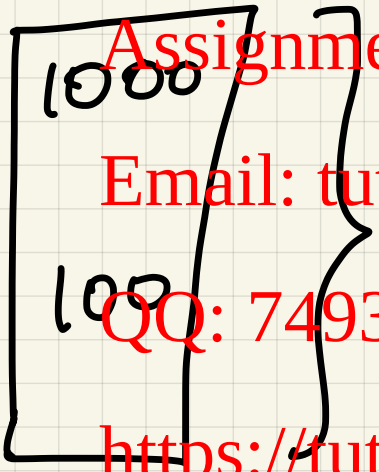
$x =$

$$\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

$$Ax = \begin{pmatrix} 1\cdot1 + 2\cdot2 + 1\cdot3 \\ 1\cdot1 + 1\cdot2 + 3\cdot1 \end{pmatrix} = \begin{pmatrix} 8 \\ 6 \end{pmatrix}$$

Back to diet problem:

$A =$ 

$b =$ 

x = vector of variables

$= \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$

So we can write constraints as

$Ax \geq b$

each element of Ax is $\geq$ corresponding element of b.

objective function can be written as

$\min \; c^T x$, where

$c^T = (1.5, 2.0, 1.2)$ = cost vector

↑ cost of 1 Mt Dew

Overall:

$$\left.\begin{array}{l} \min \; c^T x \\ \text{s.t.} \quad Ax \geq b \\ \qquad x \geq 0 \end{array}\right\}$$

If we have constraints like
$\geq$ or $\leq$, we can always
rewrite constraints, so we only
have $\geq$.

"proof" by example:

$$-x_1 + x_2 \geq 2$$

$\longrightarrow$ rewrite as

$$x_1 - x_2 \geq -2$$

Difference between Linear Algebra and Linear Programming:

Linear Algebra deals with systems of linear equations.

Linear Programming deals with systems of linear inqualitees.

So LP is much more general and useful.

Blending problems

Idea: given ingredients, blend them in products with prescribed properties

Ex: Sunco oil company.

Goal: blend crude oil into gasoline.

Crude oil is 3 types, given properties:

Octane	Sulfur %
12	0.5
6	2.0
8	3.0

Available: 5000 barrells each.

purchase price: 45, 35, 25

$/barrell.

Gasoline, types (product)

required properties: octane (at least)	sulfur (at most)
10	1%
8	2%
6	1%

sales price: 70, 60, 50

$/barrell.

demand: 3000, 2000, 3000 barrells

(at least)

production cost: $4 per barrel

set up  maximize our profit!

Variables:

c_1, c_2, c_3: barrells of crude 1,2,3 bought.

q_1, q_2, q_3: barrells of gas 1,2,3 made.

Objective:

$$\max 70q_1 + 60q_2 + 50q - (45c_1 + 35c_2 + 25c_3) - 4(q_1 + q_2 + q_3)$$

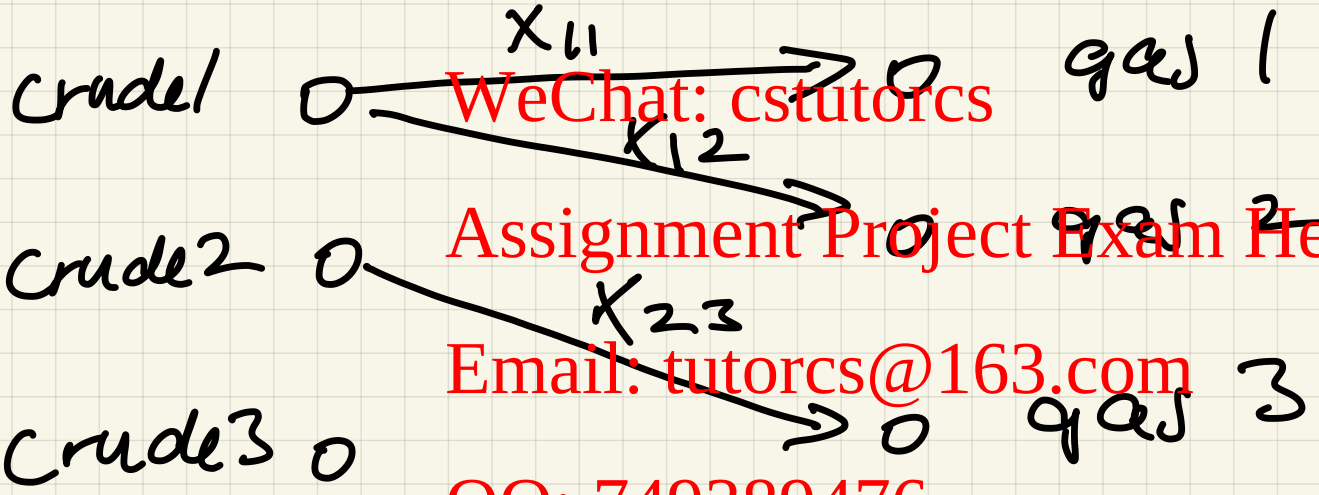
$70q_1 + 60q_2 + 50q$ → gas sale

$45c_1 + 35c_2 + 25c_3$ → price of crude

$4(q_1 + q_2 + q_3)$ → production cost

We need more variables!

x_{ij} = amount (in barrels) crude i used to make gas j.



9 new variables!

Constraints:

sulfur in gas1:

$x_{11} \cdot$ [redacted] $+ x_{21} \cdot 0.02$

Crude 1 into gas1

$+ x_{31} \cdot 0.03$

$\leq 0.01 \, g_1$

1% of gas1

similar constraints for gas 2 and gas 3.

Octane:

We assume that Octane mixes similarly

$12x_{11}$  $+ 8x_{31}$

Total Octane for gas 1

≥ 10 g1

Required Octane for gas 1 is 10.

Similar for gas 2, gas 3

octane.

Availability of Crude:

$C_1, C_2, C_3 \leq 5000$

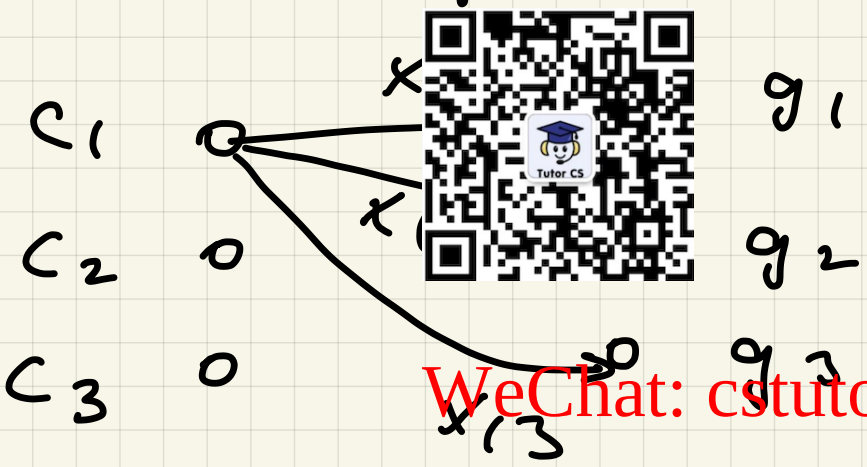
Demand for gas:

$g_1 \geq 3000, g_2 \geq 4000,$

$g_3 \geq 3000$

Connect $q_1, q_2, q_3, c_1, c_2, c_3$

to the x_{ij}.



$$\left.\begin{aligned} c_1 &= x_{11} + x_{12} + x_{13} \\ c_2 &= x_{21} + x_{22} + x_{23} \\ c_3 &= x_{31} + x_{32} + x_{33} \end{aligned}\right\}$$

Compact way of writing:

$$c_i = \sum_{j=1}^{3} x_{ij} \quad \text{for } i = 1, 2, 3.$$

Also: $q_j = \sum_{i=1}^{3} x_{ij}$ for $j = 1, 2, 3.$

Remark:

We can get rid of c_1c_2c  substituting the x_{ij} into g play where $c_1c_2c_3$ shows up.

But we do not have to expand

of $c_1c_2c_3$.

These are useful to have.

Multiperiod problems

So far everything happened at once.

We may have several periods, inventory can going from one period to the other.

Key point: draw a good picture.

Ex. (GKP, 1.2.1)

KW Oil company.

buys oil from a supplier to satisfy local demand.

Month	1	2	3	4
demand (1000 liters)	5	7	9	6
purchase price $/liter	0.75	0.72	0.92	0.90

best time to buy

storage tank can hold at most 4000 liters.

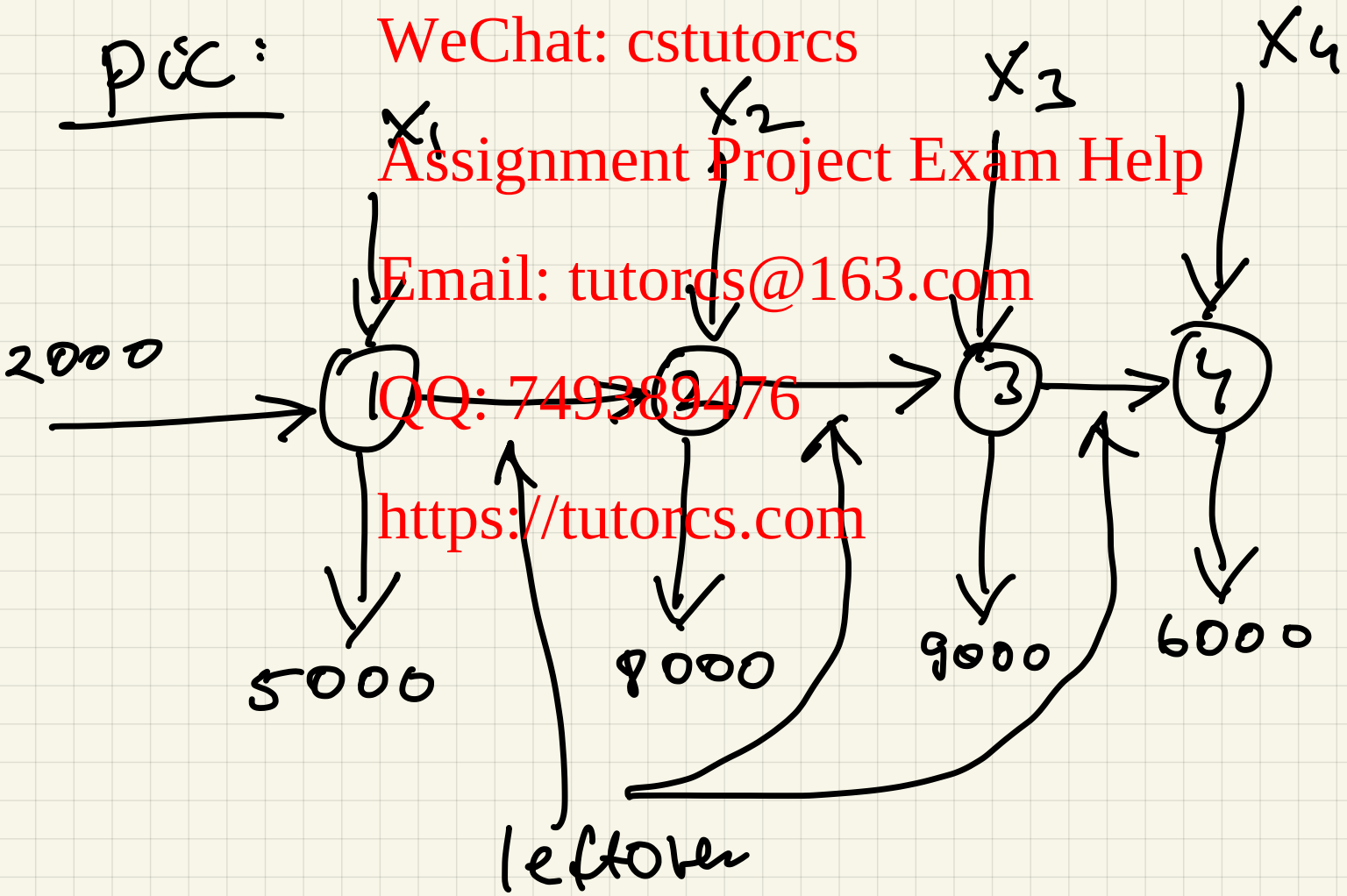
At start we have 2000 liters.

Write LP to satisfy demand exactly at minimum cost.

Variables:

x_i = amount of oil purchased in month i ($i = 1, \ldots, 4$), in barrels.

pic:



Other variables:

s_i = amount of oil left in tank at end of month i ($i = 1, 2, 3, 4$)

Objective:

Minimize

$0.75 x_2 + 0.92 x_3$

$+ 0.$

Constraints:

$2000 + x_1 \geq 5000$

Rewrite as

$2000 + x_1 = 5000 + s_1$

(balance constraint:

at a node (month),

what comes in = what goes out.

2nd month balance:

$$s_1 + x_2 = 9000 + s_2$$

3rd month balance:

$$s_2 + x_3 = 9000 + s_3$$

4th month balance:

$$s_3 + x_4 = 6000 + s_4 .$$

All variables are ≥ 0.

4th month again:

$$s_3 + x_4 = 6000 + s_4$$

$$s_i \leq 4000 \quad (i = 1, 2, 3, 4)$$

Reminder: One major application of LP is workforce management.

Technical problem

Company has following demand for skilled workers (SW)

Month	1	2	3	4
Demand	50	63	65	55
per person wages (thousand $)	3	3.1	3.2	3.4

We start with 50 SW

At end of each month, 10% of SW quit.

On 1st of each month, we accept trainees.

In 1 month, trainee → SW

After that, she stays for at least one more month.

Ex: trainee is hired on Jan 1

→ she becomes SW by Feb 1, may quit on March 1.

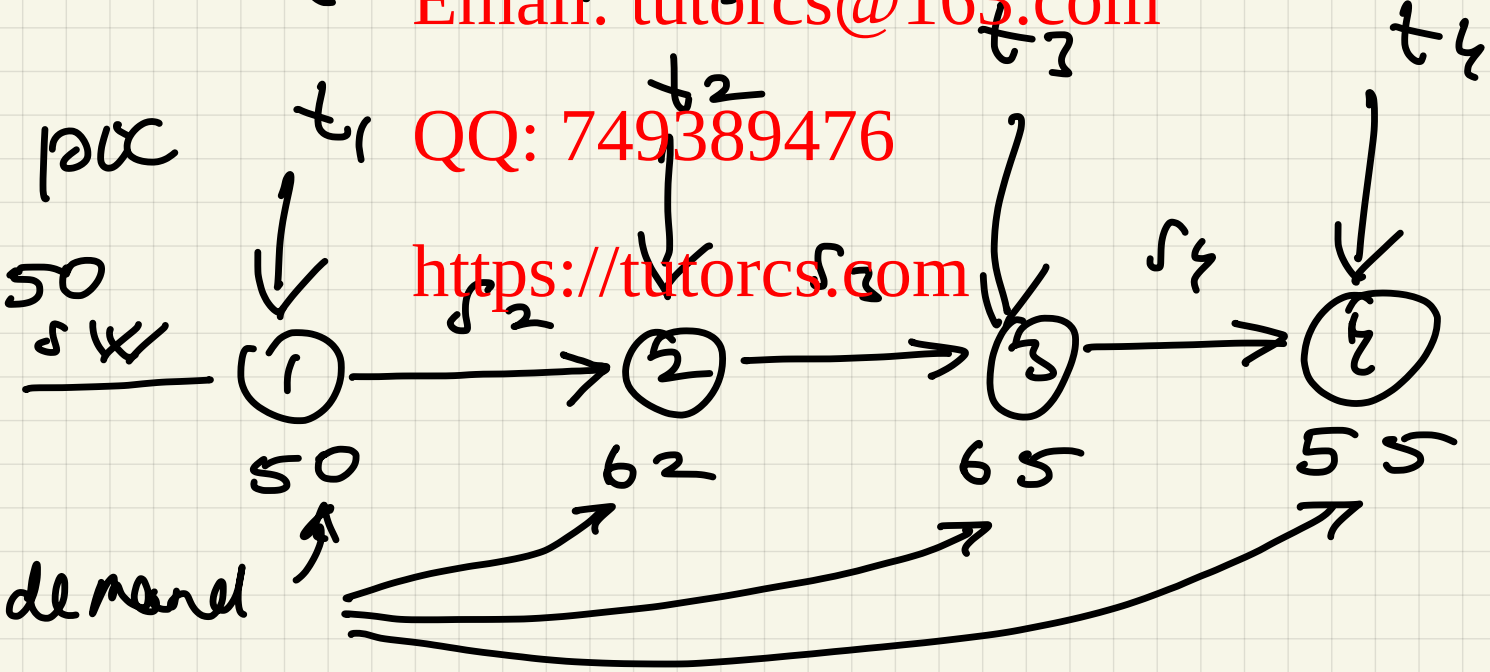
Trainees make 1K per month.

Set up LP to minimize total wages paid!

Variables:

t_i = # of trainees hired on 1st day of month i $i = 1,2,3,4$.

s_i = # of skilled workers on 1st day of month i $i = 1,2,3,4$



Objective:

$$\min \; 3 s_1 + 3.1 s_2 + 3.2 s_3$$

$$+ \text{[illegible] of SW}$$

$$+ \sum_{i=1}^{4} t_i$$

Constraints:

$$s_1 = 50$$

$$s_2 = s_1 \cdot 0.9 + t_1$$

$$s_3 = s_2 \cdot 0.9 + t_2$$

$$s_4 = s_3 \cdot 0.9 + t_3$$

$$s_2 \geq 62, \; s_3 \geq 65, \; s_4 \geq 55$$

All t_i are ≥ 0.

Suppose I have n months.

Demand for SW in month i

is d_i

At start have 50 SW.

Compact form of constraints:

$s_1 = 50$

$s_i = s_{i-1} + 0.95 s_{i-1}$

for $i = 2, \ldots, n$.

Reminder:

Linear Algebra deals with linear equations.

LP deals with linear inequalities.

Linear inequalities are much more general than linear equations.

Ex: (1) $x_1 + x_2 = 1$ is an equation. Can be rewritten as two inequalities:

$$x_1 + x_2 \leq 1 \quad (2)$$

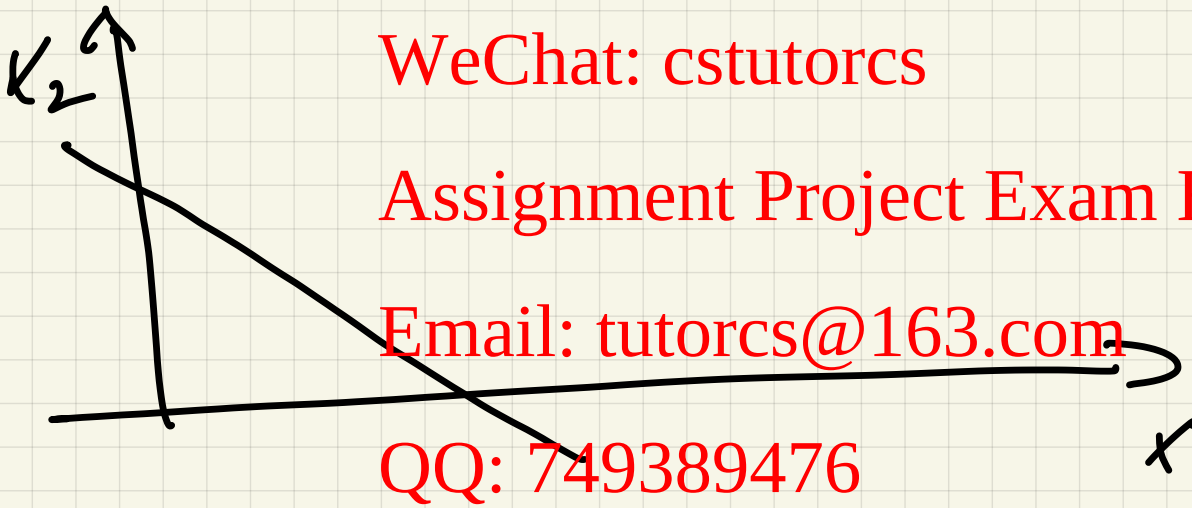
and $$x_1 + x_2 \geq 1 \quad (3)$$

(3) can be rewritten as

$$-x_1 - x_2 \leq -1$$

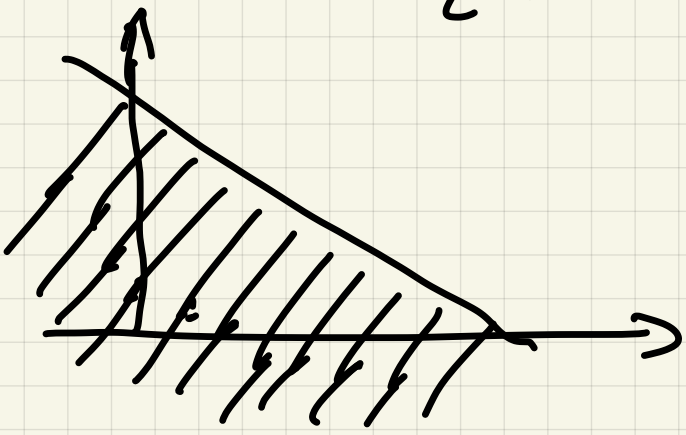
But: $x_1 + x_2 \leq 1$ cannot be written as an equality!

Geometrically:



Picture of (x_1, x_2): $x_1 + x_2 = 1$ straight line.

Picture $S = \{(x_1, x_2) : x_1 + x_2 \leq 1\}$



Some terminology:

Given a LP

Min $c^T x$

s.t. $Ax \geq b$

$x \geq 0$

A feasible solution is an x that satisfies the inequalities, namely $Ax \geq b$, $x \geq 0$.

An optimal solution is an x that is feasible, and minimizes $c^T x$.

LPs "in disguise".

Optimization problems that do not look like LPs, but we can use some tricks, and still formulate them as LPs.

Ex: (GKT, sec 1.2 / problem 3)

sketch only:

Employed at a company:

Tom, Peter, Nina, Samir, ...

Demands: Tom wants ≥ 20K in salary.

Peter, Nina, Samir each want at least 5K more than Tom.

etc.

Goal: satisfy all demands, minimize total salary.

Variable: variable for each employee.

T = salary of Tom

P = ——— Peter

N = ——— Nina, etc.

Objective: Min T + P + N + ...

Constraints:

$T \geq 20$ (Tom wants at least 20 K)

$P \geq T + 5$

$N \geq T + 5$, etc.

New objective:

same constraints, but

minimize largest salary!

Under this objective, if I have 4 employees,

(19, 19, 20, 20) as salaries is better than

(10, 10, 10, 30)

Formulate as LP:

use same variables, add a new variable, say z.

Add constraints: $z \geq T$, $z \geq P$, $z \geq N$, etc.

New objective:

Min Z

keep constants about employee demands.

why does this work?

in a feasible solution:

$Z \geq P, Z \geq T, Z \geq N, \ldots$

$\Rightarrow Z \geq \max\{P, T, N, \ldots\}$

in an optimal solution

$Z =$ at least one of the salaries

$\Rightarrow Z = \max\{P, T, N, \ldots\}$

Ex:

$$\text{Min } q(x_1, x_2)$$

$$\text{s.t. } \text{linear constraints.}$$

} (*)

where $q(x_1, x_2) =$

$$\max\{x_1 - x_2,\ x_1 + 2x_2\}$$

use same trick to reduce

new variable

$$\text{Min } z$$

$$\text{s.t. } z \geq x_1 - x_2$$

$$z \geq x_1 + 2x_2$$

keep linear constraints from (*).

why this works?

in any feasible solution

$z \geq x_1 - x_2$

$z \geq x_1 + 2x_2$

$\Rightarrow z \geq \max\{x_1 - x_2, x_1 + 2x_2\}$.

in an optimal solution

$z = x_1 - x_2$

OR

$z = x_1 + 2x_2$ (maybe both).

So in an optimal solution

$z = \max\{x_1 - x_2, x_1 + 2x_2\}$

Absolute values as LPs

Ex: $|x_1|$

s.t. linear constraints.

Formulate as LP!

$$|x_1| = \begin{cases} x_1 & \text{if } x_1 \geq 0 \\ -x_1 & \text{if } x_1 < 0 \end{cases}$$

$$= \text{Max}\{x_1, -x_1\}.$$

use trick we just learned!

use new variable z.

$$\begin{aligned} \text{Min} \quad & z \\ \text{s.t.} \quad & z \geq x_1, \quad z \geq -x_1 \\ & + \text{linear constraints.} \end{aligned}$$

Ex:

Min $|x_1| + |x_2|$

s.t. linear constraints.

Create two new variables

z_1 | z_2

Min $-z_1 + z_2$

s.t. $\left.\begin{array}{l} z_1 \geq x_1 \\ z_1 \geq -x_1 \end{array}\right\} \Rightarrow z_1 \geq |x_1|$

$\left.\begin{array}{l} z_2 \geq x_2 \\ z_2 \geq -x_2 \end{array}\right\} \Rightarrow z_2 \geq |x_2|$

linear constraints.

In an optimal solution:

$z_1 = x_1$ or $z_1 = -x_1$

$\Rightarrow z_1 = |x_1|$.

Similarly $z_2 = |x_2|$.

So, formulation is correct.

Application in regression:

Reminder: $y \in \mathbb{R}^k$

L2-norm (or 2-norm) of y is

$$\|y\|_2 = \left(\sum_{i=1}^{k} y_i^2 \right)^{1/2}$$

L1-norm (or 1-norm) of y is

$$\|y\|_1 = \sum_{i=1}^{k} |y_i|$$

Ex: $y = (1, -2, 3)^T$

$\|y\|_1 = 2 + 3 = 6$

$\|y\|_2 = (\quad + 9)^{1/2} = 3.7417$

Regression:

L2-regression (least squares)

Given $Ax = b$ linear system of equations

suppose it is over determined,

# of constraints > # of variables

m n

Find x such that

$\|b - Ax\|_2$ is minimized!

Solution:

$$x = (A^T A)^{-1} A^T b$$

L1-regression (absolute deviation regression)

Find x to minimize

$$\|b - Ax\|_1$$

can be dqu by LP!

Introduce y to model $b - Ax$

Problem becomes

$$\min |y_1| + |y_2| + \dots + |y_m|$$

$$\text{s.t.} \quad b - Ax = y$$

Still not an LP!

Use previous trick!

$$\min z_1 + z_2 + \cdots + z_m$$

s.t. $\left.\begin{matrix} z_i \\ -z_i \end{matrix}\right\} \quad i = 1, \ldots, m.$

$$y = b - Ax$$

In feasible soln: $z_i \geq |y_i| \ \forall i$

in optimal soln: $z_i = |y_i| \ \forall i$

Both have pros & cons.

Ex: $A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 1 & 1 \end{bmatrix}$ $b = \begin{bmatrix} 5 \\ 5 \\ 50 \end{bmatrix}$

3rd equation is an outlier, incorrect observation.

L2-regression solution:

$x = (5.9094, 5.9094)^T$

$y = (-12.73, -12.73, 38.18)^T$

L1-regression solution:

$x = (5/3, 5/3)^T$

$y = (0, 0, -49.35)^T$

better in detecting outliers.

Portfolio optimization

n stocks

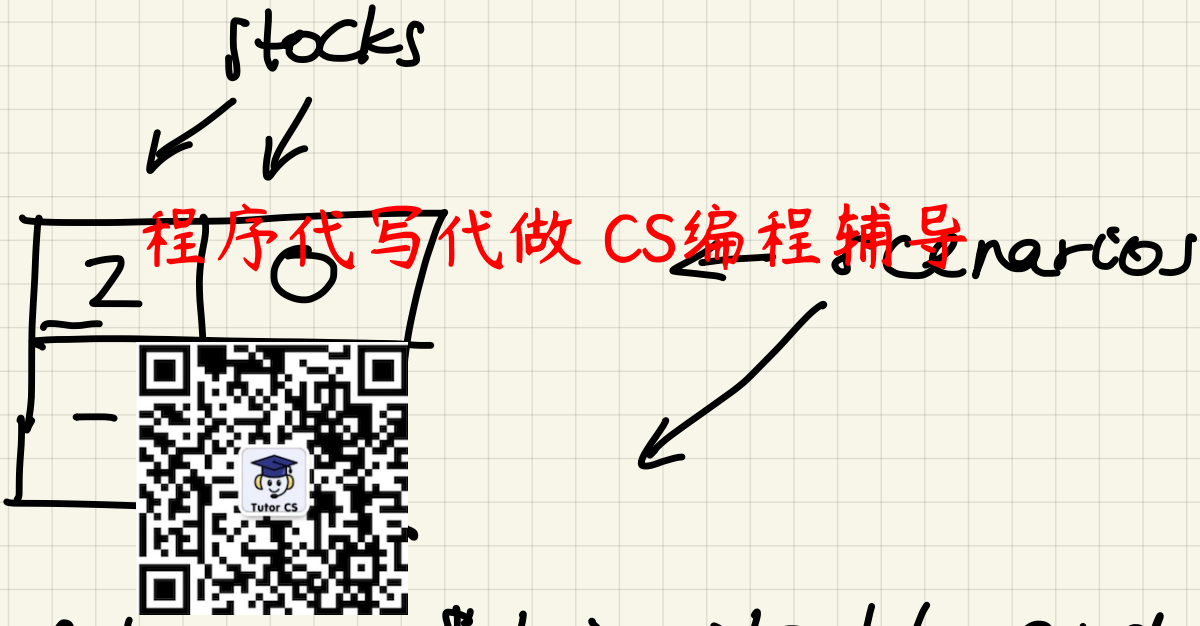
m sce  : e.g. who wins the election.

R = return matrix

Rij = return on $1 invested in stock j, if scenario i happens.

Goal: capture "return, risk", optimize using LP!

Ex:



E.g. if I put $1 in stock1 and scenario 1 happens, then I make $2.

If scenario 2 happens, then I lose $1.

p_1 = probability of scenario 1.

p_2 = ——"——————— 2.

$p_1 = 0.8$, $p_2 = 0.2$

Where should I invest?

Investment normalized:

$x \in \mathbb{R}^n, \quad x \geq 0$

$\sum_{j=1}^{n} x_j = 1$

(I assume I have $1 to invest).

In example if I invest x_1, x_2,

return in scenario 1 is

$$2x_1 + 0 \cdot x_2 =: r_1$$

in scenario 2 is

$$-x_1 + 0.1\, x_2 =: r_2$$

Return in general:

In scenario i

$$r_i := \sum_j R_{ij} x_j$$

In matrix form:

define $r \in \mathbb{R}^m$, and set

$$r = Rx.$$

First objective:

Maximize expected return!

Variables: x_j = investment in stock j $(j = 1, \dots, n)$

r_i = return in scenario i $(i = 1, \dots, m)$

Objective:

$$\text{Max} \sum_{i=1}^{m} p_i r_i$$

(p_i: probability of scenario i)

Constraints:

$$\sum_{j=1}^{n} x_j = 1$$

$B \times$

return matrix (data)

$x \geq 0.$

In example:

$$\begin{aligned} \max \quad & 0.8 r_1 + 0.2 r_2 \\ \text{s.t.} \quad & 2 x_1 + x_2 = r_1 \\ & x_1 + x_2 = 1 \\ & x_1 \geq 0, \; x_2 \geq 0 \end{aligned}$$

Optimal solution:

$(1, 0)$, i.e. $x_1 = 1, x_2 = 0$.

Catch: with probability 0.2, I lose all my money!

Terminology:

$\sum_{i=1}^{m} p_i r_i = : r_p$ expected return.

Quan: risk:

Downside risk in scenario i:

$$d_i := \begin{cases} 0, & \text{if } r_i \geq 0 \\ -r_i, & \text{if } r_i < 0 \end{cases}$$

$$= \max\{0, -r_i\}.$$

ADR = Average Downside Risk

$$:= \sum_{i=1}^{m} p_i d_i$$

Ex: $x_1 = 1,\ x_2 = 0$.

$r_1 =$ 

$r_2 =$

$$r_p = 0.8 \cdot 2 + 0.2 \cdot (-1) = 1.4$$

$$P = \begin{array}{c} \begin{matrix} x_1 & x_2 \end{matrix} \\ \begin{bmatrix} 2 & 0 \\ -1 & 0.1 \end{bmatrix} \end{array}$$

$d_1 = 0$

$d_2 = 1$

$$ADR = 0.8 \cdot 0 + 0.2 \cdot 1 = 0.2$$

2nd objective:

minimize risk!

Variable: r (as before) = downside risk in scenario i ($i = 1 \cdots m$)

Objective:

$$\text{Min} \sum_{i=1}^{m} p_i d_i$$

Constraints:

$$Rx = r$$

$$x \geq 0$$

$$\sum_{j=1}^{n} x_j = 1$$

$$d_i = \text{Max}\{0, -r_i\}$$

$i = 1, \cdots, m$.

This is NOT a LP !!

Claim: $d_i = \max\{0, -r_i\}$

can be modeled as

$$\left.\begin{array}{l} d_i \geq 0 \\ d_i \geq -r_i \end{array}\right\} \text{ for all } i.$$

Proof: In any feasible solution $d_i \geq 0$, $d_i \geq -r_i$

$$\Rightarrow d_i \geq \max\{0, -r_i\}.$$

In an optimal solution

$d_i = 0$ or $d_i = -r_i$,

otherwise I could make d_i smaller, and improve objective ($p_i > 0$).

optimal solution:

$$P = \begin{bmatrix} 2 & 0 \\ -1 & 0.1 \end{bmatrix}$$

$x_1 = 0, x_2 = 1$

Summary: two extremes

	rp (return)	ADR
(1, 0)	1.4	0.2
(0, 1)	0.02	0

(1, 0): high risk, high return

(0, 1): low risk, low return.

Balancing the two

$\sigma > 0$ parameter

Min r_p

s.t. $p \geq \sigma$

In detail, LP is:

Variables: x, r, d

Objective: $\min \sum_{i=1}^{m} p_i d_i \; (= ADR)$

Constraints:

$$Rx = r$$

$$\sum_j x_j = 1$$

$$d_i \geq -r_i \quad \forall i$$

$$r_p = \sum_i p_i r_i \geq \sigma$$

$$d_i \geq 0 \quad \forall i$$

$$x_j \geq 0 \quad \forall j$$

Solve for increasing σ_1

plot

(σ_1, μ_2 with $r_p \geq \sigma$)

σ	Min ADR
0	0
⋮	⋮
0.1	0
⋮	
1.0	0.13
⋮	
1.4	0.2

What to do with the table!

Give options to investor.

piecewise linear function.

efficient frontier

For given return, we get minimum risk

Investor can pick his/her preference of (δ, μ).

Extra: AMPL tutorial

4 steps of modeling & solving with AMPL.

Step 1: write LP on paper.

- variables
- objective
- constraints

Step 2: "walk before you run".

Write AMPL code ONLY with mod file. See steel-simple.mod

Almost the same as what we wrote on paper.

Step 3 : If separate model and data file is needed :

Data ≡ parameter

≡ ".m"

Given by user, we cannot change it.

Must be declared in model file.

Actual value appears in data file.

param must be declared before variables in model file.

Decision ≡ variables

ONLY in model file.

Step 4 : Solve LP.

Check whether the solution makes sense:

1) does it satisfy constraints?

2) can I explain my solution to a "user"?

Ex 1: basic LP

step 1: $\max 3x_1 + 2x_2$

s.t. $4x_1 + 4x_2 \leq 4$

$2x_1 + x_2 \leq 2$

$x_1 \geq 0,\ x_2 \geq 0$

step 2: basic-simple.mod

word-by-word translation of this LP.

Step 3 LP in general.

$$\max \quad c^T x$$

$$\text{s.t.} \quad \leq b$$

$$\geq 0$$

where

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & & & \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix} \qquad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

$$c^T = \begin{bmatrix} c_1 & c_2 & \cdots & c_n \end{bmatrix}$$

Constraint i in LP:

$$A_{i1} x_1 + A_{i2} x_2 + \cdots + A_{in} x_n \leq b_i$$

$$\Leftrightarrow$$

$$\sum_{j=1}^{n} A_{ij} x_j \leq b_i$$

Match the "abstract" setup with the "concrete" setup.

with p  data:

$$A = \begin{bmatrix} 1 & 4 \\ 2 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

$$c^T = \begin{bmatrix} 3 & 2 \end{bmatrix}$$

Remark: $c^T x = \sum_{j=1}^{n} c_j x_j$

$A_{11} = 1$, $A_{12} = 4$

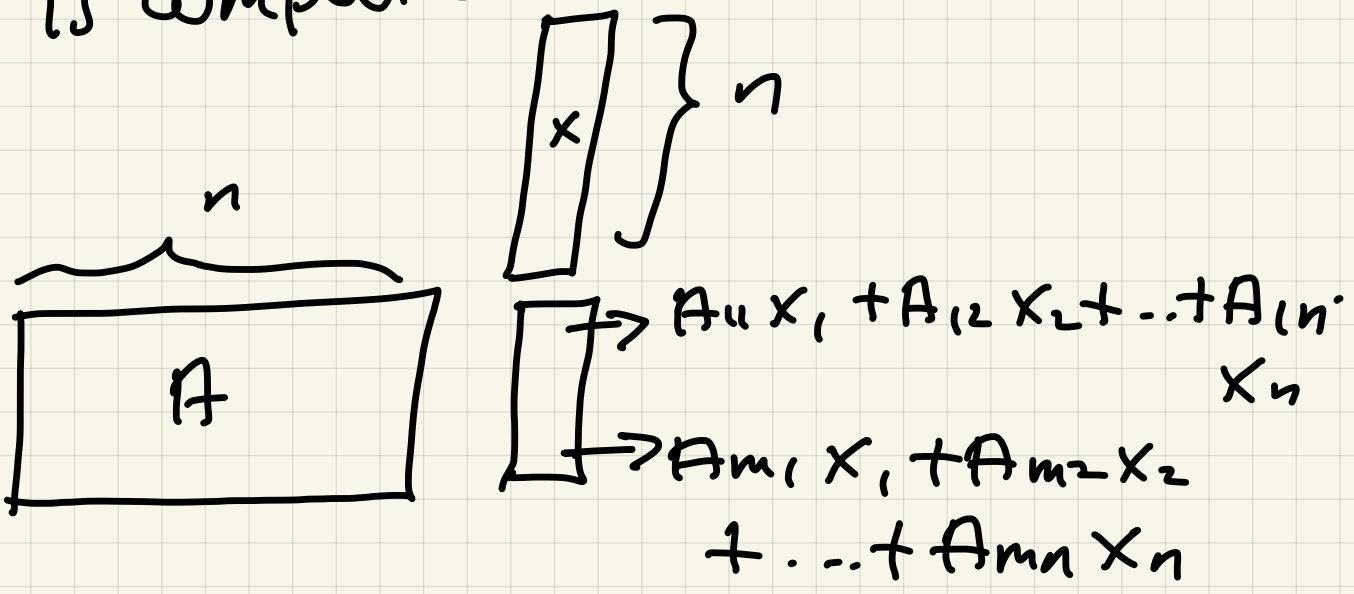
$A_{21} = 2$, $A_{22} = 1$

Matrix-vector multiplication:

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_2 & & \cdots & A_{2n} \\ & & & \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Ax is computed as



Think it over several times:

1st component of Ax is:

$$A_{11}x_1 + A_{12}x_2 + \dots + A_{1n}x_n$$

$$= \sum_{j=1}^{n} a_{1j}x_j$$

Do the same for 2nd component of Ax:

$$\begin{bmatrix} A_{21}x_1 + A_{22}x_2 + \dots + A_{2n}x_n \end{bmatrix}$$

Same for ith component of Ax

$\rightarrow$ think it over.

$$\rightarrow \sum_{j=1}^{n} a_{2j}x_j$$

Try it with numbers:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \qquad x = \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix}$$

$$Ax = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} \rightarrow \begin{bmatrix} 1\cdot 1 + 2\cdot 3 + 3\cdot 2 \\ 4\cdot 1 + 5\cdot 3 + 6\cdot 2 \end{bmatrix}$$

AMPL cont'd

Rem of steps:

1) Write [illegible] on paper

2) Write using only mod [illegible] (steel-simple.mod)

3) separate model and data file.

Ex2 (artificial)

Rem: $F_1 = 1, F_2 = 1$ (*)

$F_i = F_{i-1} + F_{i-2}$ (*)

Fibonacci numbers.

Step 1: write it as LP.

variables: $F_1, F_2, \ldots, F_n$

objective: Max F_n

constraints: (*)

Step 2: fib-simple.mod

Step 3: Fibonacci numbers in general.

$F_1 = F_2 = 1$, $F_i = F_{i-1} + F_{i-2}$ $(i = 3, 4, \ldots, n)$

In LP format

Variables: F_i $(i = 1, \ldots, n)$

Objective: Max F_n

Constraints: $F_2 = 1$

$F_i = F_{i-1} + F_{i-2}$

$(i = 3, 4, \ldots, n)$

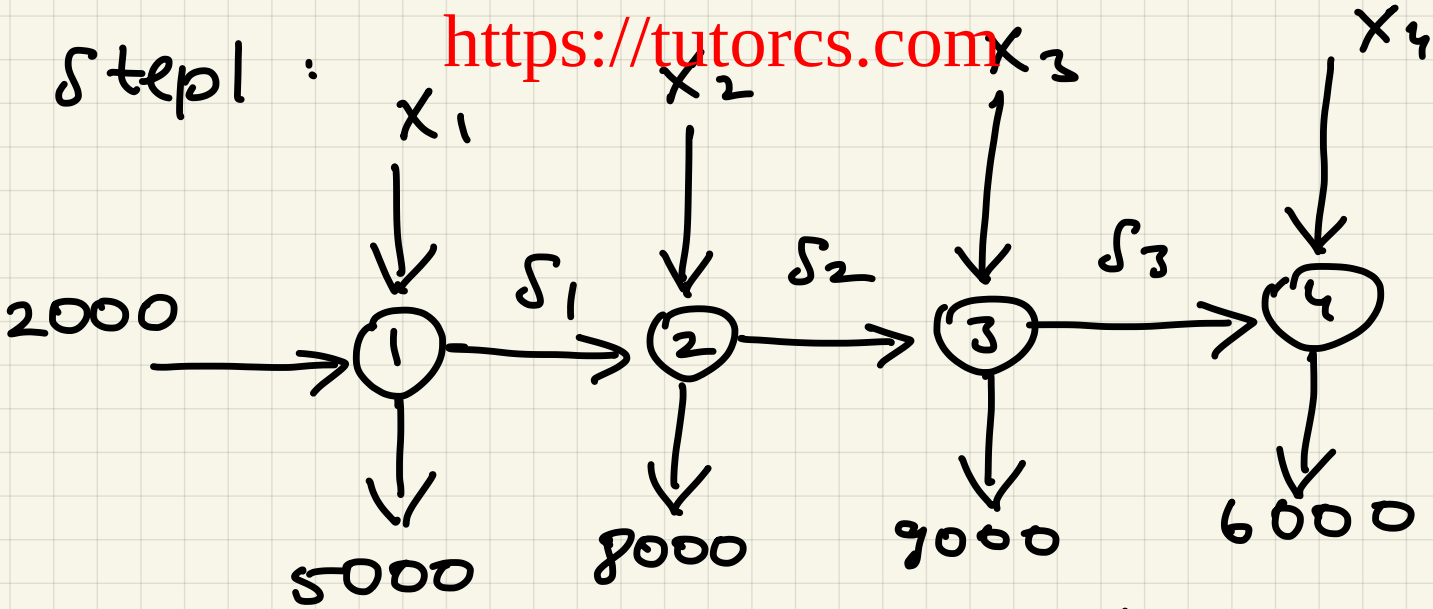
Files: fib.mod, fib.dat

KWDil

Ex 3

(oil supply to satisfy demand).

Step1:



x_i = amount purchased in month i

s_i = amount left after

We satisfied demand in month i

Objective

Min $0.7 x_1 + 0.72 x_2 + 0.92 x_3 + 0.9 x_4$

Constraints:

$2000 + x_1 = 5000 + s_1$

$s_1 + x_2 = 8000 + s_2$

$s_2 + x_3 = 9000 + s_3$

$s_3 + x_4 = 6000$

step2: KW-simple.mod

All variables ≥ 0.

Step 3 :

Data is 2000, 5000 i ... , 6000

demand

rting

ventory

0.75, 0.72, 0.92, 0.9

price

assume n months. // price in month i

$$\begin{cases} \text{Min} \quad \sum_{i=1}^{n} c_i x_i \\ \text{s.t.} \quad s_{i-1} + x_i = s_i + d_i \quad (d_i: \text{demand}) \\ \quad (i = 1, 2, \dots, n) \\ \quad s_0 = 2000 \\ \quad x_1, \dots, x_n \geq 0 \\ \quad s_0, s_1, \dots, s_n \geq 0 \end{cases}$$

End of LP modeling!

Exam (Sept 23):

only modeling.

hws through 8.

prepare: read all hws,

quizzes,

makeup quizzes.

Also AMPL: converting AMPL code.