Lecture 2

Linear Programming: Examples, Formulations, Software

Qiong Wang

Department of Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign

August 28 - August 30, 2023

Linear Programming (LP) in Standard Form

$$
\begin{aligned}
\max_{x_1,\dots,x_n} \quad & \{c_1 x_1 + ... + c_n x_n\} \\
\text{subject to} \quad & a_{11}x_1 + ... + a_{1n}x_n \le b_1, \\
& .... \\
& a_{k1}x_1 + ... + a_{kn}x_n \le b_k, \\
& x_1 \ge 0, ..., x_n \ge 0.
\end{aligned}
$$

- all decision variables take non-negative real values;
- the objective is to maximize a linear function of decision variables;
- all constraints are linear inequalities of decision variables;

  variables are on the LHS and the constant is on the RHS

  LHS $\le$ RHS.

Corporate Financing Problem

resort

short-term cash flow:

Month	Jan	Feb	Mar	Apr	May	Jun
Net cash flow	-150	-100	200	-200	-50	300

accumulated balance:

Month	Jan	Feb	Mar	Apr	May	Jun
cash balance	-150	-250	-50	-250	-300	0

outside financing is needed: credit line or commercial paper.

Short-term Financing: Credit Line

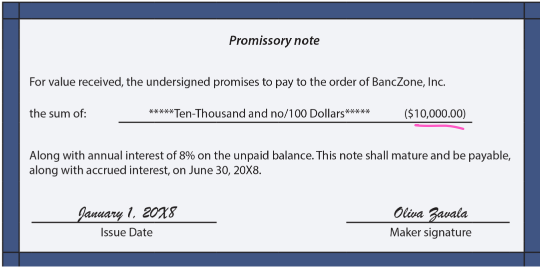
r: interest rate,
L: limit

month	balance	interest	net change
1	x_1		x_1
2	x_2	rx_1	$x_2 - x_1$
...	...	...	...
t	x_t	rx_{t-1}	$x_t - x_{t-1}$
...	...	...	...

- $x_t - x_{t-1} > 0$: borrow, more cash and interest payment
- $x_t - x_{t-1} < 0$: return, less cash and interest payment.
- creditline limit: $x_t \leq L$ for all t.

Short-term Financing: Commercial Papers

Not paying interest every month but we get discount



Promissory note

For value received, the undersigned promises to pay to the order of BancZone, Inc.

the sum of: *****Ten-Thousand and no/100 Dollars***** ($10,000.00)

Along with annual interest of 8% on the unpaid balance. This note shall mature and be payable, along with accrued interest, on June 30, 20X8.

January 1, 20X8
Issue Date

Oliva Zavala
Maker signature

- issuing date (τ_i): 01/01/2008
- amount (y_i): \$ 10,000
- maturity date (m_i): 06/30/2008
- interest rate (δ_i): 8%

Cash Position

short-term, cash flow: f_t $(t = 1, \cdots T)$.

- start from a "clean slate", the net cash position in month 1

$$z_1 = x_1 + \sum_{i:\tau_i=1} y_i + f_1,$$

- in month 2,

interest rate on credit

issue commercial paper

operating income cost

$$z_2 = z_1 + (x_2 - x_1) - rx_1 + \sum_{i:\tau_i=2} y_i - \sum_{i:m_i=2} (1+\delta_i)y_i + f_2,$$

change

2nd month

nominal value

- similarly, in any other month t ($z_t \geq 0$ for all t to avoid backruptcy).

$$z_t = z_{t-1} + (x_t - x_{t-1}) - rx_{t-1} + \sum_{i:\tau_i=t} y_i - \sum_{i:m_i=t} (1+\delta_i)y_i + f_t.$$

pay all my credit card debt

- the net value in the end $(x_T = 0)$:

$$V = z_T - \sum_{i:m_i>T} (1+\delta_i)y_i.$$

Linear Programming Formulation

$$\max_{x_t, y_i, z_t, V} \{V\}$$

subject to:

$$x_1 + \sum_{i:\tau_i=1} y_i - z_1 = -f_1,$$

$$z_{t-1} + x_t - (1+r)x_{t-1} + \sum_{i:\tau_i=t} y_i - \sum_{i:m_i=t} (1+\delta_i)y_i - z_t = -f_t, \quad t = 2, ..., T$$

$$V = z_T - \sum_{i:m_i>T} (1+\delta_i)y_i$$

$$x_T = 0, \quad x_t \le L, \quad x_t \ge 0, \quad y_i \ge 0, \quad z_t \ge 0, \quad t = 1, ..., T$$

if the surplus cash also generates interest income with rate r', then:

ก้อนใน account generate income ได้

ใส่ interest rate

replace z_{t-1} with $(1+r')z_{t-1}$, $t = 1, \cdots, T$ in the above.

ปกติ $\boxed{r' \le r}$ ไม่งั้น arbitrage → ยืมมาเก็บหมด

Example (C&T)

Month	Jan	Feb	Mar	Apr	May	Jun
cash flow	-150	-100	200	-200	-50	300

credit line: interest $r = 1\%$ per month, limit, $L = 100$;

commercial paper $(i = 1, 2, 3, (\tau_i, m_i, \delta_i))$: $(1, 4, 2\%)$, $(2, 5, 2\%)$, $(3, 6, 2\%)$.

surplus cash generates an income of 0.3% per month

$$\max_{x_t, y_i, z_t, V} \{V\} \quad \text{(in this case, } V = z_6\text{)}$$

credit line

borrow x_2 ⊖ ตัวจ่ายต้น

จ่าย interest

subject to:

$$
\begin{aligned}
x_1 + y_1 - z_1 &= 150 \\
1.003z_1 + x_2 + y_2 - 1.01x_1 - z_2 &= 100, \\
1.003z_2 + x_3 + y_3 - 1.01x_2 - z_3 &= -200, \\
1.003z_3 + x_4 - 1.01x_3 - 1.02y_1 - z_4 &= 200 \\
1.003z_4 + x_5 - 1.01x_4 - 1.02y_2 - z_5 &= 50 \\
1.003z_5 - 1.01x_5 - 1.02y_3 - V &= -300 \\
x_t \le 100,\ x_t \ge 0\ y_i \ge 0\ z_t \ge 0, &\quad i = 1, 2, 3, t = 1, \cdots, 6.
\end{aligned}
$$

Related Problem: Financing Long-term Project

- project lasts from year 1 to year T, expenses L_t $(t = 1, \cdots, T)$.
- a set of bonds $(i = 1, \cdots, n)$ available now, face value 100.
  price: p_i, coupon rate: c_i, maturity date: m_i
- fund available to cover expenses, the target is to minimize the spending (by using income from the bond).

z_t: cash in year t (z_0: initial investment, no new cash injection)

x_i: the amount of bond i invested $(i = 1, \cdots, n)$.

$$\min_{x_i, z_t} \left\{ z_0 + \sum_{i=1}^{n} p_i x_i \right\}$$

$$\text{subject to:} \quad z_{t-1} + \sum_{i:m_i \geq t} c_i x_i + \sum_{i:m_i = t} 100 x_i - z_t = L_t$$

$$x_i \geq 0 \ (i = 1, n), \quad z_t \geq 0 \ (t = 0, \cdots, T).$$

A Portfolio Management Problem

n assets with historical returns $R_i(t)$ $(1 \le i \le n)$, $t = 1, ..., T$.

estimated rate of return of each asset

$$r_i = \frac{R_i(1) + ... + R_i(T)}{T}, \quad 1 \le i \le n.$$

Problem: invest a fixed investment budget into these assets.

- decision: x_i be % invested in asset i $(i = 1, .., n)$,
- objective:

$$\max_{x_1,\ldots,x_n} \{r_1x_1 + r_2x_2 + ... + r_nx_n\}$$

- constraints:

$$x_1 + x_2 + ... + x_n \le 1, \quad \text{and} \quad x_1 \ge 0,\ x_2 \ge 0, ...,\ x_n \ge 0.$$

and more importantly

$$\text{Risk} \le \text{some threshold}$$

MAD as a Risk Measure

- average return over T years: $r_1 x_1 + ... + r_n x_n$
- actual return in year t: $x_1 R_1(t) + x_2 R_2(t) + ... + x_n R_n(t)$, where

$$\frac{1}{T}\sum_{t=1}^{T}\left(\sum_{i=1}^{n} R_i(t) x_i\right) = \sum_{i=1}^{n} r_i x_i$$

i.e.,

$$\frac{1}{T}\sum_{t=1}^{T}\left(\sum_{i=1}^{n}(R_i(t) - r_i) x_i\right) = 0$$

yearly deviation from the mean

$$\sum_{i=1}^{n}(R_i(t) - r_i) x_i \quad t = 1, \cdots T.$$

- Mean Absolute Deviation (MAD) as the risk measure:

$$\frac{1}{T}\sum_{t=1}^{T}\left|\sum_{i=1}^{n}(R_i(t) - r_i) x_i\right|$$

Mean-Risk Model

let $\bar{m}$ be the upper bound of the MAD that you can accept:

$$\max_{x_1,\ldots,x_n} \{r_1x_1 + \cdots + r_nx_n\}$$

subject to

$$x_1 + x_2 + ... + x_n \leq 1,$$

$$\frac{1}{T}\sum_{t=1}^{T}\left|\sum_{i=1}^{n}(R_i(t) - r_i)x_i\right| \leq \bar{m}.$$

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

not a linear function

issue:

$$\sum_{t=1}^{T}\left|\sum_{i=1}^{n}(R_i(t) - r_i)x_i\right|$$

is not an linear function of $x_1, \cdots, x_n$.

but, in this case, we can transform the problem and solve it as a LP.

Transformation Technique

a is a number and x is a variable, let

$$x \geq a \quad \text{and} \quad x \geq -a.$$

then

$$\left.\begin{array}{lll} a \geq 0 & \longrightarrow & x \geq a \geq -a. \\ a \leq 0 & \longrightarrow & x \geq -a \geq a. \end{array}\right\} \longrightarrow \quad x \geq |a|$$

similarly, introduce variables z_t $(t = 1, \cdots, T)$, and let

$$\left.\begin{array}{r} z_t \geq \sum_{i=1}^{n} (R_i(t) - r_i)x_i \\ \text{and} \quad z_t \geq -\sum_{i=1}^{n} (R_i(t) - r_i)x_i. \end{array}\right\}$$

add a linear constraint

equivalent

then

$$z_t \geq \left| \sum_{i=1}^{n} (R_i(t) - r_i)x_i \right| \quad t = 1, \cdots, T.$$

Solving the Problem as an LP

$$\max_{\hat{x}_1,\cdots,\hat{x}_n} \{r_1\hat{x}_1 + \cdots + r_n\hat{x}_n\}$$

subject to:

$$\hat{x}_1 + \cdots + \hat{x}_n \leq 1,$$

$$\hat{x}_1 \geq 0, \cdots, \hat{x}_n \geq 0,$$

$$\cancel{\frac{1}{T}\sum_{t=1}^{T}\left|\sum_{i=1}^{n}(R_i(t) - r_i)\hat{x}_i\right| \leq \bar{m},}$$

$$\frac{1}{T}(z_1 + \cdots + z_T) \leq \bar{m},$$

$$z_t \geq \sum_{i=1}^{n}(R_i(t) - r_i)\hat{x}_i \;\; (t = 1, ..T),$$

$$z_t \geq -\sum_{i=1}^{n}(R_i(t) - r_i)\hat{x}_i \;\; (t = 1, .., T).$$

This is an LP, and we will show it is equivalent to the original problem by making two comparisons.

Comparison 1

$$\Pi_1 = \max_{x_1,\cdots,x_n} \{r_1 x_1 + \cdots + r_n x_n\}$$

$$\sum_{i=1}^{n} x_i \le 1, \quad x_1 \ge 0, \cdots, x_n \ge 0,$$

$$\frac{1}{T}\sum_{t=1}^{T}\left|\sum_{i=1}^{n}(R_i(t) - r_i)x_i\right| \le \bar{m}.$$

$$\Pi_2 = \max_{\hat{x}_1,\cdots,\hat{x}_n} \{r_1 \hat{x}_1 + \cdots + r_n \hat{x}_n\}$$

$$\sum_{i=1}^{n} \hat{x}_i \le 1, \quad \hat{x}_1 \ge 0, \cdots, \hat{x}_n \ge 0,$$

$$\frac{1}{T}(z_1 + \cdots + z_T) \le \bar{m},$$

$$z_t \ge \sum_{i=1}^{n}(R_i(t) - r_i)\hat{x}_i \ (t = 1, ..T),$$

$$z_t \ge -\sum_{i=1}^{n}(R_i(t) - r_i)\hat{x}_i \ (t = 1, .., T).$$

If $(x_1^*, \cdots, x_n^*)$ is the optimal solution to problem 1, then

$$\hat{x}_1 = x_1^*, \cdots, \hat{x}_n = x_n^* \quad z_t = \left|\sum_{i=1}^{n}(R_i(t) - r_i)x_i^*\right| \ (t = 1, \cdots, T)$$

is a solution to problem 2, so $\Pi_1 \le \Pi_2$.

Comparison 2

nonlinear

$$\Pi_1 = \max_{x_1,\cdots,x_n} \{r_1 x_1 + \cdots + r_n x_n\}$$

$$\sum_{i=1}^{n} x_i \leq 1, \quad x_1 \geq 0, \cdots, x_n \geq 0,$$

$$\frac{1}{T}\sum_{t=1}^{T}\left|\sum_{i=1}^{n}(R_i(t) - r_i)x_i\right| \leq \bar{m}.$$

linear problem

$$\Pi_2 = \max_{\hat{x}_1,\cdots,\hat{x}_n} \{r_1 \hat{x}_1 + \cdots + r_n \hat{x}_n\}$$

$$\sum_{i=1}^{n} \hat{x}_i \leq 1, \quad \hat{x}_1 \geq 0, \cdots, \hat{x}_n \geq 0,$$

$$\frac{1}{T}(z_1 + \cdots + z_T) \leq \bar{m},$$

$$z_t \geq \sum_{i=1}^{n}(R_i(t) - r_i)\hat{x}_i \ (t = 1, ..T),$$

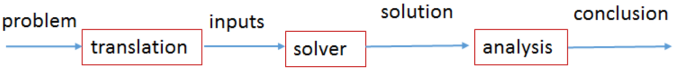
$$z_t \geq -\sum_{i=1}^{n}(R_i(t) - r_i)\hat{x}_i \ (t = 1, .., T).$$

If $(\hat{x}_1^*, \cdots, \hat{x}_n^*, z_1^*, \cdots, z_T^*)$ is the optimal solution to problem 2, then

$$\frac{1}{T}\sum_{t=1}^{T}\left|\sum_{i=1}^{n}(R_i(t) - r_i)\hat{x}_i^*\right| \leq \frac{1}{T}\sum_{t=1}^{T} z_t^* \leq \bar{m},$$

so $(\hat{x}_1^*, \cdots, \hat{x}_n^*)$ is a solution to problem 1, and $\Pi_1 \geq \Pi_2$.

Optimization Solver and Modeling Language



LP solvers: avoid repetitive developments and use best expertise available. we will use gurobi (https://www.gurobi.com/)

AMPL: a modeling language that facilitates translation and analysis

- teaching copy (incl. gurobi for Windows or MacOS) on Canvas.
- online resources: http://ampl.com/
- book: http://ampl.com/resources/the-ampl-book/chapter-downloads/

Using AMPL

teaching copy in "software" folder on Canvas course site, expires on Dec 31.

installation guide:
https://ampl.com/try-ampl/ampl-for-courses/ampl-course-install/

Windows: download ampl_mswin64 from Canvas

- extract files to a program folder (e.g., you may named the folder ampl).
- create a user folder for your files and programs.
- create an ampl program and store it in the user folder.
- locate ampl.exe or amplide.exe and run.

Mac OS X: download ampl_macos64, click ampl and then amplide. If amplide.exe does not run because it is not from Mac App store (google to find a solution):

- macOS Catalina and macOS Mojave:
  1. System Preferences > Security & Privacy,
  2. under the General tab. Click Open Anyway to confirm your intent to open or install the app.
  3. you can open it in the future by double-clicking it.
- older macOS: control-click the program icon and choose open.
- if it says the file is damaged, reboot the machine.

Problem Example

A fund wants to allocate no more that \$1m into two bonds:

- corporate bond: yield 4%, 3-year maturity, and rating A (score 2).
- government bond: yield 3%, 4-year maturity, and rating Aaa (score 1).

Question: how to maximize the yield, while keeping the average maturity below 3.6 years and the rating score no more than 1.2.

LP Formulation:
x_1: investment in corporate bond,
x_2: investment in government bond.

$$
\begin{aligned}
& \max_{x_1, x_2} \{4x_1 + 3x_2\} \\
\text{subject to:} \quad & 3x_1 + 4x_2 \leq 3.6, \\
& 2x_1 + x_2 \leq 1.2, \\
& x_1 + x_2 \leq 1, \\
& x_1 \geq 0, x_2 \geq 0.
\end{aligned}
$$

Example: Direct Translation

model (translate into a model file)

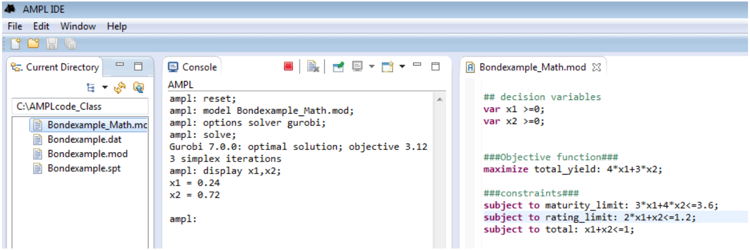
$$\max_{x_1, x_2} \{4x_1 + 3x_2\}$$

subject to:

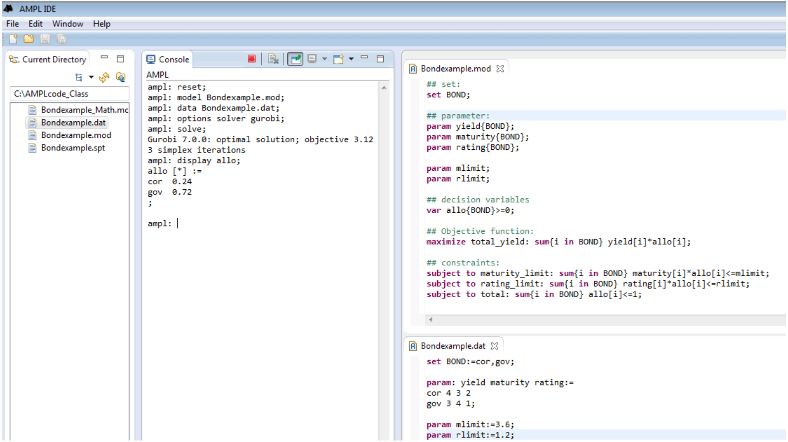
$$
\begin{aligned}
3x_1 + 4x_2 &\le 3.6, \\
2x_1 + x_2 &\le 1.2, \\
x_1 + x_2 &\le 1, \\
x_1 \ge 0, x_2 &\ge 0.
\end{aligned}
$$

steps:

1. reset to start
2. set path and load model
3. choose solver
4. solve
5. display outputs

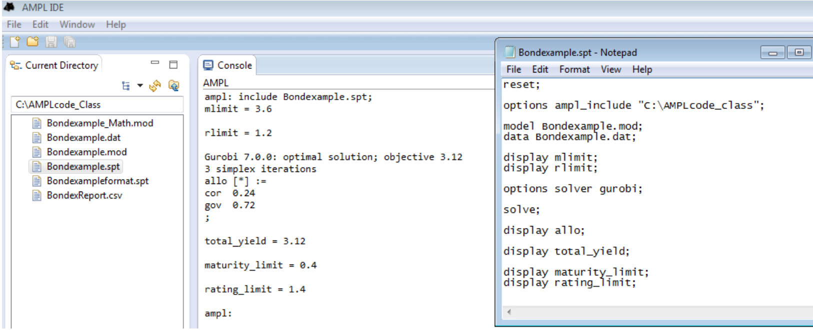


Example: Better Implementation, Reusable Model



- separate model and data into different files;
- impose structure and naming in the model: set, parameter, variable, obj., and constraints.

Example: Use of Command Script



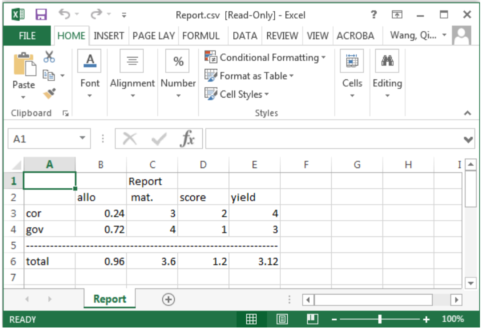
- set path;
- load model and data files;
- display input parameters;
- choose the solver and solve;
- display outcomes.

What Else Can You Do with a Script


```
reset;

options ampl_include "C:\AMPLcode_class";

model Bondexample.mod;
data Bondexample.dat;

options solver gurobi;

solve;

printf "\n\n----Optimization Output ----------\n";
printf "opt. obj:\t%0.3f\n",total_yield;

printf "----------------------------------\n";

printf "\n\n\tmat.\tscore\tyield\t|opt_allo.\n";

printf "-------------------------------|-------\n";
for {i in BOND} {
        printf "%s\t%0.2f\t%0.2f\t%0.2f\t|%0.2f\n",
                i,maturity[i],rating[i],yield[i],allo[i];
}


printf ",,Report, \n,allo, mat.,score,yield\n" >Report.csv;
for {i in BOND} {
        printf "%s,%0.2f,%0.2f,%0.2f,%0.2f\n",
                i,allo[i], maturity[i],rating[i],yield[i]>Report.csv;
}
printf "-----------------------------------------------------\n"> Report.csv;
printf "total,%0.2f,%0.2f,%0.2f,%0.2f\n", sum{i in BOND} allo[i],
                sum{i in BOND} allo[i]*maturity[i],
                sum{i in BOND} allo[i]*rating[i],
                sum{i in BOND} allo[i]*yield[i] > Report.csv;
```

Screen and File Outputs


```
ampl: include Bondexampleformat.spt;
Gurobi 7.0.0: optimal solution; objective 3.12
3 simplex iterations


----Optimization Output ----------
opt. obj:       3.120
----------------------------------


        mat.    score   yield   |opt_allo.
--------------------------------|-------
cor     3.00    2.00    4.00    |0.24
gov     4.00    1.00    3.00    |0.72
ampl:
```



	A	B	C	D	E
1			Report		
2		allo	mat.	score	yield
3	cor	0.24	3	2	4
4	gov	0.72	4	1	3
5	--				
6	total	0.96	3.6	1.2	3.12
7					

What Else Can You Do with a Script: Analysis

what happens to the optimal allocation, what is the optimal yield, when the constraint on the rating changes:

Bondexampleiterate.spt - Notepad

```
reset;

options ampl_include "C:\AMPLcode_class";

model Bondexample.mod;
data Bondexample.dat;

options solver gurobi;

printf "rating,cor, gov, yield\n" > Report2.csv;

for{r in 1..8}{

        let rlimit:=1+0.1*r;

        solve;

        printf "%0.2f,%0.2f,%0.2f,%0.2f\n",
                rlimit,allo["cor"],allo["gov"],total_yield
                        >Report2.csv
}
```

report2.csv

rating	cor	gov	yield
1.1	0.16	0.78	2.98
1.2	0.24	0.72	3.12
1.3	0.32	0.66	3.26
1.4	0.4	0.6	3.4
1.5	0.5	0.5	3.5
1.6	0.6	0.4	3.6
1.7	0.7	0.3	3.7
1.8	0.8	0.2	3.8