IEOR E4007
T. Lin

Homework # 1

Instructions

- Submit a single PDF file of your solutions and codes (EXCEL + python notebook) onto `Gradescope`
- Submit EXCEL workbooks and python notebooks onto the `Assignment` section of Courseworks.

1. *Exercise 3.2 in Tutuncu + Cornuejols text*

   A company will face the following cash requirements in the next eight quarters (positive entries represent cash needs while negative entries represent cash surpluses).

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
100	500	100	-600	-500	200	600	-900

   The company has the following products available:

   - A 2-year, i.e. 8 quarter, loan available *only* at the beginning of Q1, with a 1% interest per quarter, compounded quarterly. This loan has to be repaid at the *beginning* of Q9.
   - A 6-month, i.e. 2 quarter, loan with 1.8% interest per quarter, compounded quarterly. Available at the beginning of Q1, . . . , Q7.
   - A quarterly loan with a 2.5% interest for the quarter. Available at the beginning of Q1, . . . , Q8.
   - Any surplus can be invested at a 0.5% interest per quarter.

   Formulate a linear program that maximizes the wealth of the company at the beginning of Q9. Solve this problem using python **and** EXCEL.

   Let

   - $q_t \geq 0$: amount borrowed using quarterly loan with interest rate $r_q = 2.5\%$, $t = 1, \ldots, T-1$.
   - $s_t \geq 0$: amount borrowed using six-month loan with interest rate $r_s = 1.8\%$, $t = 1, \ldots, T-2$.
   - $l_1 \geq 0$: amount borrowed using the long-term loan in quarter 1 with interest rate $r_l = 1\%$.
   - $z_t \geq 0$: excess cash available in period $t = 1, \ldots, T$.

The inflow − outflow = 0 constraints are as follows:

$$
\begin{array}{rcll}
(l_1 + q_1 + s_1) - z_1 & = & c_1 & \\
(q_2 + s_2 + (1+r_f)z_1) - ((1+r_q)q_1 + z_2) & = & c_2 & \\
(q_t + s_t + (1+r_f)z_{t-1}) - ((1+r_q)q_{t-1} + (1+r_s)^2 s_{t-2} + z_t) & = & c_t, & t \in \{3, \ldots, 7\}, \\
(q_8 + (1+r_f)z_7) - ((1+r_q)q_7 + (1+r_s)^2 s_6 + z_8) & = & c_8, & \\
((1+r_f)z_8) - ((1+r_l)^8 l_1 + (1+r_q)q_8 + (1+r_s)^2 s_7 + z_9) & = & 0, &
\end{array}
$$

Here we are assuming that the interest is *compounded* quarterly.

The objective is:

$$\max z_9$$

The optimal value for this problem is 471.56, as shown in the softwares.

```
opt. value
  471.5631

         l           q           s           z
  399.8099           0           0    299.8099
                     0    198.6910           0
              100.0000           0           0
                     0           0    291.5917
                     0           0    793.0497
                     0           0    597.0149
                     0           0           0
                     0                900.0000
                                      471.5631
```

2. *Bond arbitrage problem*

Suppose the price structure of three bonds is as follows:

Bond	Current price	Expiration	Coupon
1	101.625	10/15/2023	6.875
2	101.5625	4/15/2024	5.50
3	103.80	4/15/2024	7.750

The term *coupon* refers to the interest rate paid on the face value of the bond (in this case, take the face value to 100 for all three bonds). The coupon is usually paid every six months and last coupon is paid on maturity. Also, on maturity the holder is paid the face value of the bond. Thus, Bond 1 will pay a coupon of 3.4375 on 4/15 and 10/15 of every year until maturity, i.e. there is only one coupon remaining. Thus, on 10/15/2023, Bond 1 pays 100 + 3.4375 = 103.4375.

Given the current price structure, the question is whether one can make an infinite amount of money. To answer this we look for an *arbitrage*. An arbitrage exists if there exists a combination of bond sales and purchases today that yields

- a positive cash flow today
- non-negative cash flows at all future dates

Since an arbitrage will allow investors to make an infinite amount of money, we expect that the bond prices at any point in time will be set so that no arbitrage opportunities exist.

(a) Formulate a linear program that identifies an arbitrage opportunity if it exists. Write a python code to identify this arbitrage opportunity.

Suppose one were to hold a position $\mathbf{x} = (x_1, x_2, x_3)^T$ in the three bonds. On 10/15/2023, we require the cash flow to be positive, i.e.

$$103.4375x_1 + 2.75x_2 + 3.875x_3 \geq 0.$$

And similarly, on 4/15/2024 we require

$$102.75x_2 + 103.875x_3 \geq 0.$$

In terms of the objective, one could maxmize the total in-flow. But in the presence of an arbtrage this would be infinite – to get a bounded solution constrain $|x_i| \leq 1$. So, now the LP becomes

$$\begin{aligned} \text{maximize} \quad & -101.625x_1 - 101.5625x_2 - 103.80x_3 \\ \text{subject to} \quad & 103.4375x_1 + 2.75x_2 + 3.875x_3 \geq 0 \\ & 102.75x_2 + 103.875x_3 \geq 0 \\ & -1 \leq x_i \leq 1, \quad i = 1, \ldots, 3. \end{aligned}$$

If this LP has a positive value, we have an arbitrage opportunity. On solving this LP using Python we get

$$\mathbf{x} = \begin{bmatrix} 0.0105 \\ 1.0000 \\ -0.9892 \end{bmatrix}$$

Thus, we have an arbitrage opportunity.

(b) Usually bonds are bought at an *ask price* and sold at a *bid price*. Consider the same three bonds listed in the table above and suppose the bid/ask prices are as listed in the table below. Show that these prices do not admit any arbitrage.

Bond	Ask price	Bid Price
1	101.6563	101.5938
2	101.5938	101.5313
3	103.7813	103.7188

In this problem one will have to account for buying and selling separately because of the difference in the bid/ask price.

Formulate a linear program that identifies an arbitrage opportunity if it exists. Write a python code to identify the arbitrage opportunity.

Let u_i denote the amount of bond i bought and v_i denote the amount of bond i sold. Then, the initial in-flow is given by

$$-101.6563u_1 + 101.5938v_1 - 101.5938u_2 + 101.5313v_2 - 103.7813u_3 + 103.7188v_3.$$

And the cash flows:

$$\begin{aligned} 103.4375(u_1 - v_1) + 2.75(u_2 - v_2) + 3.875(u_3 - v_3) &\geq 0, \\ 102.75(u_2 - v_2) + 103.875(u_3 - v_3) &\geq 0. \end{aligned}$$

Therefore, argueing as in part (a), we get the LP to solve look for arbitrage opprtunities is:

$$\begin{array}{rl} \text{maximize} & -101.6563u_1 + 101.5938v_1 - 101.5938u_2 + 101.5313v_2 - 103.7813u_3 + 103.7188v_3 \\ \text{subject to} & 103.4375(u_1 - v_1) + 2.75(u_2 - v_2) + 3.875(u_3 - v_3) \geq 0, \\ & 102.75(u_2 - v_2) + 103.875(u_3 - v_3) \geq 0, \\ & 0 \leq u_i, v_i \leq 1, i = 1, \ldots, 3. \end{array}$$

If this LP has a positive value, we have an arbitrage opportunity. On solving this LP using Python we get $(u_i, v_i) = (0, 0)$, $i = 1, 2, 3$ to machine precision. So there is no arbitrage opportunity.