Convex-Optimization

1) There are three food items available, corn, milk, and bread. The table contains, the cost per serving, the amount of Vitamin A per serving, and the number of calories per serving for each food item. Also, there are restrictions on the total number of calories (between 2000 and 2250) and the total amount of Vitamin A (between 5000 and 50,000) intake in the diet.The goal of the diet problem is to select a set of food items that will satisfy a set of daily nutritional requirement at minimum cost. The maximum number of servings is 10 per food item. Table 1 shows all the content per serving.

Food	Cost	Vitamin A	calories
Corn	0.18 USD	107	72
Milk	0.23 USD	500	121
Wheat Bread	0.05 USD	0	65

TABLE I

Solution:
Consider,

Description	Parameter	Value
Cost per serving	$\mathbf{c}$	$\begin{pmatrix}0.18\\0.23\\0.05\end{pmatrix}$
Vitamin A per serving	$\mathbf{v}$	$\begin{pmatrix}107\\500\\0\end{pmatrix}$
Calories per serving	$\mathbf{u}$	$\begin{pmatrix}72\\121\\65\end{pmatrix}$
No. of servings	$\mathbf{x}$	$\begin{pmatrix}x\\y\\z\end{pmatrix}$

TABLE II

Objective function

$$z = \min_{\mathbf{x}} \mathbf{c}^T \mathbf{x} \tag{1}$$

Constraints

$$5000 \le 107x + 500y + 0z \le 50000 \tag{2}$$
$$2000 \le 72x + 121y + 65z \le 2250 \tag{3}$$
$$0 \le x \le 10 \tag{4}$$
$$0 \le y \le 10 \tag{5}$$
$$0 \le z \le 10 \tag{6}$$

Writing all the constraints in the matrix form

$$\mathbf{px} = \mathbf{q} \tag{7}$$

$$\begin{pmatrix} 107 & 500 & 0 \\ -107 & -500 & 0 \\ 72 & 121 & 65 \\ -72 & -121 & -65 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 50000 \\ -5000 \\ 2250 \\ -2000 \\ 10 \\ 10 \\ 10 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{8}$$

By providing the objective function and constraints to cvxpy, we get the optimal cost (z) and optimal no.of servings ($\mathbf{x}$).
cvxpy code,

```
https://github.com/gadepall/EE5606−
    optimization/codes/opt_1.py
```

From cvxpy, we get

$$z = 3.16\text{USD} \tag{9}$$

$$\mathbf{x} = \begin{pmatrix} 2 \\ 10 \\ 10 \end{pmatrix} \tag{10}$$

2) Funding an expense stream. Your task is to fund an expense stream over n time periods. We consider an expense stream $\mathbf{e} \in R^n$, so that e_t is our expenditure at time t. One possibility for funding the expense stream is through our bank account. At time period t, the account has balance b_t and we withdraw an amount w_t. The value of our bank account accumulates with an

interest rate ρ per time period, less withdrawals

$$b_{t+1} = (1+\rho)b_t - w_t$$

We assume the account value must be nonnegative, so that $b_t \geq 0$ for all t. We can also use other investments to fund our expense stream, which we purchase at the initial time period t = 1, and which pay out over the n time periods. The amount each investment type pays out over the n time periods is given by the payout matrix **P**, defined so that P_{tj} is the amount investment type j pays out at time period t per dollar invested. There are m investment types, and we purchase $x_j \geq 0$ dollars of investment type j. In time period t, the total payout of all investments purchased is therefore given by $(Px)_t$. In each time period, the sum of the withdrawals and the investment payouts must cover the expense stream, so that

$$w_t + (Px)_t \geq e_t$$

for all t = 1, . . . , n. The total amount we invest to fund the expense stream is the sum of the initial account balance, and the sum total of the investments purchased: $b_1 + 1^T\mathbf{x}$.
Using the data in **expense stream data.*****,**Show that the minimum initial investment that funds the expense stream can be found by solving a convex optimization problem.
Solution:
Consider,

Description	Parameter	Value
Payout Matrix	$\mathbf{P}_{nxm}$	from expense stream data.py
expense stream	$\mathbf{e}_{nx1}$	from expense stream data.py
Interest rate	ρ	from expense stream data.py
No. of investments	m	from expense stream data.py
Time period	n	from expense stream data.py
Present Bank balance	$\mathbf{b}_{nx1}$	?
Bank withdrawals	$\mathbf{w}_{nx1}$	?
Investments purchased	$\mathbf{x}_{mx1}$	?
Total payable for investments purchaced	$\mathbf{Px}_{nx1}$	?

TABLE III

Given to minimize the inital investment.

Objective Function,

$$Z = \min_{b1,\mathbf{x}} b_1 + 1^T\mathbf{x} \tag{11}$$

Constraints,

$$\mathbf{b} \geq 0 \tag{12}$$
$$\mathbf{x} \geq 0 \tag{13}$$
$$\mathbf{w} + \mathbf{Px} \geq \mathbf{e} \tag{14}$$
$$b_{t+1} = (1+\rho)b_t - w_t. \tag{15}$$

Solution of the above objective function with the constraints is obtained by CVXPY.
CVXPY code,

https://github.com/gadepall/EE5606–optimization/codes/opt_2.py

https://github.com/gadepall/EE5606–optimization/codes/expense_stream _data.py

The optimal initial investment Z = 177.51

3) We are tasked with designing a box shaped with width w, height h, and depth d. We are given that the total wall area is at most 200 square units, the total floor area is at most 60 square units, and the aspect ratios (h/w ans d/w) are at least 0.8 and at most 1.2. Formulate an optimization program to solve for the dimensions h,w and d that results in a box of the largest possible volume, and implement in CVXPY.
Solution:
Given

Description	Parameter	Value
Width	w	?
Height	h	?
Depth	d	?
Volume	v	?
Wall area	2(wh+hd)	$\leq$ 200
Floor area	dw	$\leq$ 60
Aspect ratio = h/w and d/w	a_1 and a_2	$\leq$ 1.2 and $\geq$0.8

TABLE IV

Let **w** be the optimization variables vector.

$$\mathbf{w} = \begin{pmatrix} w \\ h \\ d \end{pmatrix} \tag{16}$$

Objective is to maximize the volume of the box.

$$v = whd$$

Objective function is defined as,

$$z = \max_{w,h,d}(whd) \tag{17}$$

constraints are,

$$wh \le 200 \tag{18}$$
$$dw \le 60 \tag{19}$$
$$0.8 \le a_1 \le 1.2 \tag{20}$$
$$0.8 \le a_2 \le 1.2 \tag{21}$$

Writing constraints in the matrix form,

$$\mathbf{w}^T\mathbf{P_1}\mathbf{w} \le q_1 \tag{22}$$
$$\begin{pmatrix} w & h & d \end{pmatrix}\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} w \\ h \\ d \end{pmatrix} \le 100 \tag{23}$$
$$\mathbf{w}^T\mathbf{P_2}\mathbf{w} \le q_2 \tag{24}$$
$$\begin{pmatrix} w & h & d \end{pmatrix}\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} w \\ h \\ d \end{pmatrix} \le 60 \tag{25}$$
$$\mathbf{P_3}\mathbf{w} \le \mathbf{q_3} \tag{26}$$
$$\begin{pmatrix} -0.8 & -1 & 0 \\ 1 & -1.2 & 0 \\ -0.8 & 0 & -1 \\ 1 & 0 & -1.2 \end{pmatrix}\begin{pmatrix} w \\ h \\ d \end{pmatrix} \le \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{27}$$

Solving the objective function and constrains by cvxpy.
cvxpy code,

```
https://github.com/gadepall/EE5606−
    optimization/codes/opt_3.py
```

The maximum optimum value is 387.29 cubic units.
Optimum w = 7.74, h = 6.45, d = 7.74 units.

4) A fictional company AccessApple & Co. produces three types of covers for Apple products: one for iPod, one for iPad, and another for iPhone. The company's production facilities are such that if we devote the entire production to iPod covers, we can produce 6000 of them in a day. If we devote the entire production to iPhone covers or iPad covers, we can produce 5000 or 3000 of them, respectively, in one day. The production schedule is one work week of 5 working days, and the week's production must be stored before distribution. Storing 1000 iPod covers (packaging included) takes up 40 cubic feet of space. Storing 1000 iPhone covers (packaging included) takes up 45 cubic feet of space, and storing 1000 iPad covers (packaging included) takes up 210 cubic feet of space. The total storage space available is 6000 cubic feet. Due to a commercial agreement, AccessApple & Co. has to deliver at least 5000 iPod covers, and 4000 iPad covers per week in order to strengthen the products' diffusion. The marketing department estimates that the weekly demand for iPod covers, iPhone, and iPad covers does not exceed 10000 and 15000 , and 8000 units, respectively. Therefore the company does not want to produce more than these numbers. Finally, the net profits in USD for each iPod cover, iPhone cover, and iPad cover are USD 4, USD 6 and USD 10 respectively. The aim is to determine a weekly production schedule that maximizes the total net profit.

Solution:
Consider,
x= proportion of time devoted each day to iPod cover production,
y= proportion of time devoted each day to iPhone cover production,
z= proportion of time devoted each day to iPad cover production.

Description	Parameter	Value
Proportion of time devoted	$\mathbf{x}$	$\begin{pmatrix} x \\ y \\ z \end{pmatrix}$
Production per day	$\mathbf{p}_p$	$\begin{pmatrix} 6,000 \\ 5,000 \\ 3,000 \end{pmatrix}$
Storage per 1000 units	$\mathbf{s}$	$\begin{pmatrix} 40 \\ 45 \\ 210 \end{pmatrix}$
minium requirement	$\mathbf{m}_r$	$\begin{pmatrix} 5,000 \\ 0 \\ 4,000 \end{pmatrix}$
Maximum Production	$\mathbf{m}_p$	$\begin{pmatrix} 10,000 \\ 15,000 \\ 8,000 \end{pmatrix}$
Profit	$\mathbf{p}$	$\begin{pmatrix} 4 \\ 6 \\ 10 \end{pmatrix}$
profit for week	$\mathbf{c}$	$\begin{pmatrix} 120000 \\ 15000 \\ 15000 \end{pmatrix}$

TABLE V

The elements of **c** are the product of production per day, profit and no.of workings day in a week for ipod, iphone and ipad respectively.
Objective function,

$$z = \max_{\mathbf{x}} \mathbf{c}^T \mathbf{x} \tag{28}$$

$$z = \min_{\mathbf{x}} (-\mathbf{c}^T \mathbf{x}) \tag{29}$$

constrains,

$$x + y + z \leq 1 \tag{30}$$

$$1200x + 1125y + 3150 \leq 6000 \tag{31}$$

$$30,0000x \geq 5000 \tag{32}$$

$$15,000z \geq 4000 \tag{33}$$

$$30,000x \leq 10,000 \tag{34}$$

$$25,000y \leq 15,000 \tag{35}$$

$$15,000z \leq 8,000 \tag{36}$$

$$\mathbf{x} \geq 0 \tag{37}$$

Putting all the constrains in the form$\mathbf{P}\mathbf{x} = \mathbf{q}$,

$$\begin{pmatrix} 1 & 1 & 1 \\ 1200 & 1125 & 3150 \\ -30,000 & 0 & 0 \\ 0 & 0 & -15,000 \\ 30,000 & 0 & 0 \\ 0 & 25,000 & 0 \\ 0 & 0 & 15,000 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \mathbf{x} \leq \begin{pmatrix} 1 \\ 6,000 \\ -5,000 \\ -4,000 \\ 10,000 \\ 15,000 \\ 8,000 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{38}$$

Solving the objective function and constrains by cvxpy.
cvxpy code,

https://github.com/gadepall/EE5606–optimization/codes/opt_4.py

The maximum weekly profit= 1,45,000 USD.

Optimum time devoted, $\mathbf{x} = \begin{pmatrix} 0.1667 \\ 0.4129 \\ 0.4203 \end{pmatrix}$.