LP & QP Assignment

Jiaxuan Zhang, Yiting Li

$E1 = 9, E2 = 13, E3 = 5$

Using measuerments_physical.csv for Q3, measuerments for Q4

07 October 2020

Contents

1 Question 1 **2**
1.1 Quesiont1.(a) 2
1.1.1 Assumption 2
1.1.2 Analysis 2
1.1.3 Model 2
1.2 Question1.(b) 2
1.2.1 Solution 3
1.2.2 Answer 3
1.3 Question 1(c) 3
1.3.1 Assumption 3
1.3.2 Analysis 4
1.3.3 Model 4
1.3.4 Solution 4
1.3.5 Answer 5

2 Question 2 **5**
2.1 Analysis 5
2.2 Solution 6
2.3 Answer 6

3 Question 3 **6**
3.1 Analysis 6
3.2 Model 7
3.3 solution 7

4 Question 4 **7**
4.1 Analysis 7
4.2 Model 9
4.3 Solution 10
4.4 Answer 10

MATLAB Code **11**

1 Question 1

1.1 Quesiont1.(a)

1.1.1 Assumption

A maximum linear programming problem was described in this question. We assume that:

1. We will install central air conditioner X, the number of which is x, and split-type air conditioner Y, the number of which is y.
2. All air conditioners are installed at one time.
3. Although x and y are integer, we consider them as non-negative real number first.

1.1.2 Analysis

Apparently, there are two constrains, namely, budget constrains and amount constrains.

1. We can not install more than 12 air conditioners or install a negative number of air conditioners.
2. Total budget for all installation is €$24000 + 300E1$, where $E1 = 9$.

1.1.3 Model

According to analysis above, we can formalize this optimization problem, we have:

$$\begin{aligned} \max_{x,y} \quad & 4x + 2.5y \\ s.t. \quad & x + y \leq 12 \\ & 3000x + 1500y \leq 24000 + 300E1 \\ & x, y \geq 0 \end{aligned} \tag{1}$$

Obviously, model above isn't a standard form of linear programming problem. The corresponding standard form is as follows:

$$\begin{aligned} -\min_{x,y,s_1,s_2} \quad & -4x - 2.5y \\ s.t. \quad & x + y + s_1 = 12 \\ & 3000x + 1500y + s_2 = 24000 + 300E1 \\ & x,\ y,\ s_1,\ s_2 = 0 \end{aligned} \tag{2}$$

Formula (14) is a standard form of LP problem described in Question1.(a).

1.2 Question1.(b)

When solving problem in Question1.(a) by using MATLAB, we will use following function:

$$[x, val, flag] = linprog(c, A, b, A_{eq}, b_{eq}, lb, ub, options) \tag{3}$$

Formula (3) represent solution of following questions:

$$\begin{aligned} \min_{x} \quad & fX \\ AX &\leq b \\ A_{eq}X &= b_{eq} \\ lb \leq X &\leq ub \end{aligned} \tag{4}$$

In formula (3) x represent the optimization of independent variable, val represent the optimized outcome, $flag$ is a sign of whether the problem has a solution. When the $flag$ is 1, it means that the problem has an optimal solution. If the $flag$ is 0, the problem has no optimal solution. We can easily tell that, in Question1.(b):

$$\begin{gathered} f = \begin{bmatrix} -4 & -2.5 \end{bmatrix} \\ A = \begin{bmatrix} 1 & 1 \\ 3000 & 1500 \end{bmatrix} \\ b = \begin{bmatrix} 12 \\ 24000 + 300E1 \end{bmatrix} \\ lb = \begin{bmatrix} 0 & 0 \end{bmatrix} \\ ub = \begin{bmatrix} inf & inf \end{bmatrix} \\ A_{eq} = b_{eq} = [\,] \end{gathered} \tag{5}$$

1.2.1 Solution

MATLAB code, which is not shown here, will be uploaded as an attachment in the form of .m file. Without considering that the result should be integer, we should install 5.8 air conditioners X, and 6.2 air conditioners Y.

Taking integral constraints into consideration, installation plan $(x, y) = (5, 7), (x, y) = (6, 5), (x, y) = (6, 6)$ should be calculated. $(x, y) = (5, 7)$ obtain maximum power 37.5 kW, $(x, y) = (6, 5)$ obtain power 36.5 kW and they all meet the other constraints. But $(x, y) = (6, 6)$ break the budget constraints.

1.2.2 Answer

The final result is that we should install 5 air conditioners X, and 7 air conditioner Y, leading to a maximum power, which is 37.5 kW.

1.3 Question 1(c)

1.3.1 Assumption

For Question 1 (c), make the following assumptions:

1. Only the cost of maintenance and the cost of installation of air conditioners are considered.
2. All budgets are not separated, and can be used flexibly after determining the service life.
3. After the durable time is pre-determined, the practical duration for using and maintenance these air conditioners should be equal to the pre-determined durable time.

1.3.2 Analysis

According to assumption 2, after the service life is determined, part of the maintenance budget can also be used to install air conditioners. For N years, the cost for installation and maintenance during N years should be considered together as shown in the following formula.

$$\sum_{n=1}^{N} 3000x + 1500y + C_x(n)x + C_y(n)y \leq 24000 + 300E_1 + (4000 + 100E_1)N \tag{6}$$

$C_x(n)$ is the maintenance cost of year n for air conditioner X, $C_y(n)$ is the maintenance cost of year n for air conditioner Y.

1.3.3 Model

According to analysis 1.3.2, the model of Question 1(c) can be made as following (for N years).

$$\begin{aligned} &\min_{x,y} -(4x + 2.5y) \\ s.t. \quad & x + y \leq 12 \\ \sum_{n=1}^{N} 3000x + 1500y + C_x(n)x + C_y(n)y &\leq 24000 + 300E_1 + (4000 + 100E_1)N \\ & x, y \geq 0 \end{aligned} \tag{7}$$

1.3.4 Solution

Different durations lead to different amount of maintenance budget can be used as installation budget. We calculated the maximal value of different durable years separately and find the best one from them.

Based on formula in 1.3.2, the corresponding parameters in MATLAB function `linprog` are shown as follows:

$$f = \begin{bmatrix} -4 & -2.5 \end{bmatrix}$$
$$A = \begin{bmatrix} 1 & 1 \\ 3000 + \sum_{n=1}^{N} C_x(n) & 1500 + \sum_{n=1}^{N} C_y(n) \end{bmatrix}$$
$$b = \begin{bmatrix} 12 & 24000 + 300E_1 + (4000 + 100E_1)N \end{bmatrix}^T$$
$$lb = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$$

For years ranging from 1 to 10, the installation plan and the maximum power are found as shown in the following table.

duration	1	2	3	4	5
max power	41.7069	44.0789	45.8932	46.5493	47.0043
X amount	7.8046	9.3860	10.5955	11.0329	11.3362
y amount	4.1954	2.6140	1.4045	0.9671	0.6638
duration	**6**	**7**	**8**	**9**	**10**
max power	46.6992	45.8421	44.6238	43.1983	41.6791
X amount	11.1328	10.5614	9.7492	8.7989	7.7861
y amount	0.8672	1.4386	2.2508	3.2011	4.2139

Table 1: Results for different duration time (non-integrization)

Without considering the integral, The optimal choice duration is 5 year, with 11.3362 X type, 0.6638 Y type and maximum power 47.0043 (kW). The next step is to adjust this result to integer and check whether it is still the best plan.

Considering that the duration time is 5 years, calculate the power of $(x, y) = (11, 0)$ $(x, y) = (11, 1)$ and $(x, y) = (12, 0)$. gives a maximal power 46.5 (kW). However, this value is lower than the maximum power of 4 and 6 years in the table 1. This means that the integer solution of 6 need to be tested extra to find the optimal solution. See the following table for the maximal power for durable years 4,5,6.

year	4			5			6		
amount x	11	11	12	11	11	12	11	11	12
amount y	0	1	0	0	1	0	0	1	0
power	44	46.5	48	44	46.5	48	44	46.5	48
whether feasible	Y	Y	N	Y	Y	N	Y	Y	N

Table 2: Results for integer points

The results show that the maximum power of 11 air conditioner X and 1 air conditioner Y remains unchanged, which is 46.5 kW for durable years 4,5,6.

1.3.5 Answer

1. The problem is hard to transform to a sinlge LP problem.
2. The durable years can be selected as 4,5,6, 11 air conditioner X and 1 air conditioner Y should be chosen and the maximum available power is 46.5kW.

2 Question 2

2.1 Analysis

In order to transform model to a discrete-time model, we need to use the following approximation:

$$\dot{T_{b,k}} = \frac{dT_{b,k}}{dt} \approx \frac{T_{b,k+1} - T_{b,k}}{\Delta t} \tag{8}$$

Where $T_{b,k}$ represents the indoor temperature in the building at time step k.

The target equation has derivative on both sides, but according to Question 3, the derivate of $q_{solar,k}, q_{occ,k}, q_{ac,k}, q_{vent,k}$ are able to observed directly. Therefore, we only need to discretize the left side of the equation.

2.2 Solution

The process to obtain discrete-time model is shown as follows.

$$\begin{aligned}
\frac{T_{b,k+1} - T_{b,k}}{\Delta t} &= a_1 \dot{q_{solar,k}} + a_2[\dot{q_{occ,k}} + \dot{q_{ac,k}} - \dot{q_{vent,k}}] + a_3[T_{amb,k} - T_{b,k}] \\
\Rightarrow T_{b,k+1} &= \{a_1 \dot{q_{solar,k}} + a_2[\dot{q_{occ,k}} + \dot{q_{ac,k}} - \dot{q_{vent,k}}] + a_3[T_{amb,k} - T_{b,k}]\}\Delta t + T_{b,k} \\
\Rightarrow T_{b,k+1} &= \{a_1 \dot{q_{solar,k}} + a_2[\dot{q_{occ,k}} + \dot{q_{ac,k}} - \dot{q_{vent,k}}] + a_3 T_{amb,k}\}\Delta t + (1 - a_3\Delta t)T_{b,k}
\end{aligned} \tag{9}$$

so,

$$\begin{aligned}
A &= 1 - a_3\Delta t \\
B &= \begin{bmatrix} a_1\Delta t & a_2\Delta t & a_2\Delta t & -a_2\Delta t & a_3\Delta t \end{bmatrix}
\end{aligned} \tag{10}$$

2.3 Answer

From 2.2, the answer is:

$$\begin{aligned}
A &= 1 - a_3\Delta t \\
B &= \begin{bmatrix} a_1\Delta t & a_2\Delta t & a_2\Delta t & -a_2\Delta t & a_3\Delta t \end{bmatrix}
\end{aligned} \tag{11}$$

3 Question 3

3.1 Analysis

In order to have a more accurate indoor temperature discrete model, we need the best coefficient of a_1, a_2, a_3 is needed. It can be realized by minimizing the square error between real $T_{b,k+1}$ and estimation.
From 2.1.2, we can rewrite the formula:

$$\sum_{i=1}^{N} \{[(T_{b,k+1} - T_{b,K}) - \Delta t(a_1 \dot{q_{solar,k}} + a_2(\dot{q_{occ,k}} + \dot{q_{ac,k}} - \dot{q_{vent,k}}) + a_3(T_{amb,k} - T_{b,k})]\} \tag{12}$$

We set:

$$
\begin{aligned}
T &= (T_{b,k+1} - T_{b,K}) \\
(\dot{q_{occ,k}} + \dot{q_{ac,k}} - \dot{q_{vent,k}}) &= (\dot{q_{sigma,k}}) \\
T_{amb,k} - T_{b,k} &= T_{sigma,k} \\
(a_1\dot{q_{solar,k}} + a_2(\dot{q_{sigma,k}}) + a_3(T_{sigma,k}) &= \begin{bmatrix} \dot{q_{solar,k}} & \dot{q_{sigma,k}} & \dot{T_{sigma,k}} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \\
\begin{bmatrix} \dot{q_{solar,k}} & \dot{q_{sigma,k}} & \dot{T_{sigma,k}} \end{bmatrix} &= \begin{bmatrix} \dot{q_k} \end{bmatrix} \\
\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} &= \begin{bmatrix} a \end{bmatrix}
\end{aligned}
\tag{13}
$$

We can write $\sum_{i=1}^{N} \begin{bmatrix} \dot{q_k} \end{bmatrix}$ as $\begin{bmatrix} \dot{q} \end{bmatrix}$, then formula 12 can be rewritten to:

$$
\begin{aligned}
&\left[T - [\dot{q}]\ [a]\right]^T \left[T - [\dot{q}]\ [a]\right] \\
&= a^T \dot{q}^T \dot{q} a - 2T^T \dot{q} a + T^T T
\end{aligned}
\tag{14}
$$

3.2 Model

This is a quadratic programming problem, standard form of a quadratic programming is:

$$
\begin{aligned}
\min_x \quad & \frac{1}{2} x^T H x + c^T x \\
Ax &= b \\
x &\geq 0
\end{aligned}
\tag{15}
$$

For formula 14, it is a standard form of quadratic programming problem, where:

$$
\begin{aligned}
H &= 2\dot{q}^T \dot{q} \\
c^T &= 2T^T \dot{q}
\end{aligned}
\tag{16}
$$

3.3 solution

We can use function in MATLAB to solve it. In Question 3, we use data set measurements_physical.csv. Details of each matrix are store in MATLAB, as a result, we only show final result here:

$$
\begin{aligned}
a_1 &= 2.662 \times 10^{-6} \\
a_2 &= 1.056 \times 10^{-5} \\
a_3 &= 2.135 \times 10^{-5}
\end{aligned}
\tag{17}
$$

4 Question 4

4.1 Analysis

The model of the problem has been clearly stated in the problem document, so the key point is to transform the model into a matrix available in MATLAB. Define the variable vector as follows:

$$x = \begin{bmatrix} \dot{q_{ac,1}} & \dot{q_{ac,2}} & \cdots & \dot{q_{ac,2160}} & T_{b,2} & T_{b,3} & \cdots & T_{b,2160} \end{bmatrix}$$

Considering the unit of Φ_k is €/kWh and the unit of $\dot{q_{ac,k}}$ is kW, so ΔT in $\sum_{k=1}^{N} \Phi_k \dot{q_{ac,k}}$ should be $\Delta t = 1(h)$.

There are two ways for finding the corresponding matrices H and C in the standard form of quadratic optimization problem.

1. Measure 1: from equation to matrices

$$\begin{aligned}\sum_{k=1}^{N} \Phi_k \dot{q_{ac,k}} \Delta t + (0.1 + E_2/10)(T_{B,k} - T_{ref})^2 &= \sum_{k=1}^{N} \Phi_k \dot{q_{ac,k}} + \sum_{k=1}^{N} 1.4(T_k^2 + T_{ref}^2 - 2T_r T_k) \\ &= \sum_{k=1}^{N} \Phi_k \dot{q_{ac,k}} - 2.8 T_r T_k + \sum_{k=1}^{N} 1.4 T_{ref}^2 + \sum_{k=1}^{N} 1.4 T_k^2 \\ &= \sum_{k=1}^{N} 1.4 T_{ref}^2 + C^T x + \frac{1}{2} x^T H x \end{aligned} \tag{18}$$

so,

$$C = [\underbrace{\Phi_1 \ \cdots \ \cdots \ \Phi_{2160}}_{N} \ \underbrace{-2.8T_{ref} \ \cdots \ \cdots \ -2.8T_{ref}}_{N-1}]^T$$

$$H = \begin{array}{c} \begin{array}{cc} \overbrace{\hphantom{0 \ \cdots \ 0}}^{N} & \overbrace{\hphantom{0 \ \cdots \ 0}}^{N-1} \end{array} \\ \left[\begin{array}{ccc|ccc} 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \\ \hline 0 & \cdots & 0 & 2.8 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 2.8 \end{array}\right] \begin{array}{l} \left.\vphantom{\begin{array}{c}0\\ \vdots\\0\end{array}}\right\} N \\ \left.\vphantom{\begin{array}{c}0\\ \vdots\\0\end{array}}\right\} N-1 \end{array} \end{array} \tag{19}$$

2. Measure 2: directly from matrices operation

$$\begin{aligned}\sum_{k=1}^{N} \Phi_k \dot{q_{ac,k}} \Delta t + (0.1 + E_2/10)(T_{B,k} - T_{ref})^2 &= \sum_{k=1}^{N} \Phi_k \dot{q_{ac,k}} + \sum_{k=1}^{N} 1.4(T_k^2 + T_{ref}^2 - 2T_r T_k) \\ &= (\sum_{k=1}^{N} \Phi_k \dot{q_{ac,k}}) - 1.4(Ax - T_{mref})^T (Ax - T_{mref}) \\ &= (\sum_{k=1}^{N} \Phi_k \dot{q_{ac,k}}) - 1.4x^T A^T A x - 1.4 T_{mref}^T T_{mref} + 2.8 T_{mref}^T A x \\ &= -2.8 T_{mref}^T T_{mref} + \Phi x + 2.8 T_{mref}^T A x - 1.4 x A^T A x \end{aligned} \tag{20}$$

where:

$$\Phi = [\underbrace{\Phi_1 \ \cdots \ \cdots \ \Phi_{2160}}_{N} \ \underbrace{0 \ \cdots \ \cdots \ 0}_{N-1}]$$

$$T_{mref} = [\underbrace{0 \ \cdots \ \cdots \ a0}_{N} \ \underbrace{T_{ref} \ \cdots \ \cdots \ T_{ref}}_{N-1}]$$

$$A = \begin{array}{c} \begin{array}{cc} \overbrace{\qquad\qquad\quad}^{N} & \overbrace{\qquad\qquad\quad}^{N-1} \end{array} \\ \left[\begin{array}{ccc|ccc} 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \\ \hline 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 1 \end{array}\right] \end{array} \begin{array}{l} \left.\vphantom{\begin{array}{c}0\\ \vdots\\0\end{array}}\right\} N \\ \left.\vphantom{\begin{array}{c}0\\ \vdots\\0\end{array}}\right\} N-1 \end{array} \tag{21}$$

so,

$$C = \Phi + 2T_{mref}^T = [\underbrace{\Phi_1 \ \cdots \ \cdots \ \Phi_{2160}}_{N} \ \underbrace{-2.8T_{ref} \ \cdots \ \cdots \ -2.8T_{ref}}_{N-1}]^T$$

$$H = 2A^TA = \begin{array}{c} \begin{array}{cc} \overbrace{\qquad\qquad\quad}^{N} & \overbrace{\qquad\qquad\quad}^{N-1} \end{array} \\ \left[\begin{array}{ccc|ccc} 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \\ \hline 0 & \cdots & 0 & 2.8 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 2.8 \end{array}\right] \end{array} \begin{array}{l} \left.\vphantom{\begin{array}{c}0\\ \vdots\\0\end{array}}\right\} N \\ \left.\vphantom{\begin{array}{c}0\\ \vdots\\0\end{array}}\right\} N-1 \end{array} \tag{22}$$

For constraints $A_{eq}x = b_{eq}$, the matrices are shown

$$A_{eq} = \underbrace{\left[\begin{array}{cccc} a & 0 & \cdots & 0 \\ 0 & a & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a \end{array}\right.}_{N-1} \underbrace{\left|\begin{array}{c} 0 \\ \vdots \\ 0 \end{array}\right|}_{1} \underbrace{\left.\begin{array}{ccccc} 1 & 0 & \cdots & & 0 \\ -1 & \ddots & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{array}\right]}_{N-1} \tag{23}$$

$$b_{eq} = \begin{bmatrix} (a_1\dot{q_{solar,1}} + a_2\dot{q_{occ,1}}t - a_2\dot{q_{vent,1}} + a_3T_{amb,1})\Delta t + T_{b,1} \\ \vdots \\ (a_1\dot{q_{solar,N}} + a_2\dot{q_{occ,N}} - a_2\dot{q_{vent,N}} + a_3T_{amb,N})\Delta t \end{bmatrix}$$

4.2 Model

The model has been clearly explained in the problem document, and the model is explained by using matrices here. Because $1.4T_{ref}^2$ is a constant in this problem, it was ignored in the standard form, it will be added after optimization with MATLAB.

4.3 Solution

Based on equations, the corresponding parameters in MATLAB function `quadprog` are shown as follows:

$$H = \begin{array}{c} \overbrace{}^{N} \quad \overbrace{}^{N-1} \\ \left[\begin{array}{ccc|ccc} 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \\ \hline 0 & \cdots & 0 & 2.8 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 2.8 \end{array}\right] \end{array} \begin{array}{l} \left.\vphantom{\begin{array}{c}0\\ \vdots\\ 0\end{array}}\right\} N \\ \left.\vphantom{\begin{array}{c}0\\ \vdots\\ 0\end{array}}\right\} N-1 \end{array}$$

$$f = [\underbrace{\Phi_1 \ \cdots \ \cdots \ \Phi_{2160}}_{N} \ \underbrace{-2.8T_{ref} \ \cdots \ \cdots \ -2.8T_{ref}}_{N-1}]^T$$

$$A_{eq} = \left[\begin{array}{cccc|c|ccccc} & & & & & 1 & 0 & \cdots & & 0 \\ a & 0 & \cdots & 0 & & & & & & \\ & & & & 0 & -1 & \ddots & \ddots & & \vdots \\ 0 & a & \ddots & \vdots & & & & & & \\ & & & & \vdots & 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 & & & & & \\ & & & & & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a & & & & & & \\ & & & & & 0 & \cdots & 0 & -1 & 1 \end{array}\right]$$

$$\underbrace{}_{N-1} \ \underbrace{}_{1} \ \underbrace{}_{N-1}$$

$$b_{eq} = \begin{bmatrix} (a_1\dot{q}_{solar,1} + a_2\dot{q}_{occ,1}t - a_2\dot{q}_{vent,1} + a_3T_{amb,1})\Delta t + T_{b,1} \\ \vdots \\ (a_1\dot{q}_{solar,N} + a_2\dot{q}_{occ,N} - a_2\dot{q}_{vent,N} + a_3T_{amb,N})\Delta t \end{bmatrix}$$

$$lb(i) = \begin{cases} 0 & \text{i=1,...,N} \\ 15 & \text{i ¿ 2160 and } \dot{q}_{occ,(i-2160)} > 0 \\ -\inf & (others) \end{cases}$$

$$ub(i) = \begin{cases} \dot{q}_{ac,max} & \text{i=1,...,N} \\ 28 & \text{i ¿ 2160 and } \dot{q}_{occ,(i-2160)} > 0 \\ \inf & (others) \end{cases}$$

Finally, the quadprog got the optimal answer, which obtained the minimal cost of €20533.

4.4 Answer

1. The optimal cost for air-conditioning along the horizon of N(2160) steps is €20533.

MATLAB Code

```
syms E1 E2 E3;
maintenance = [(200+E2) (200+2*E2) (200+3*E2) (300+4*E2) (300+5*E2) ...
    (400+5*E2) (500+5*E2) (600+5*E2) (700+5*E2) (800+5*E2);
     (50 +E3) (50 +2*E3) (100+3*E3) (150+4*E3) (150+5*E3) (200+5*E3) ...
        (250+5*E3) (300+5*E3) (350+5*E3) (400+5*E3)];%cost of maintenance
E1=9;
E2=13;
E3=5;

line_optim_opt = optimoptions('linprog', 'Algorithm', 'dual-simplex');
quad_optim_opt = optimoptions('quadprog', 'Algorithm', ...
    'interior-point-convex');

%data for Q3
measurement3  = readtable("measurements_physical.csv");

∆_t = 3600;
Qocc3   = measurement3{(1:2160),1};
Qac3    = measurement3{(1:2160),2};
Qvent3  = measurement3{(1:2160),3};
Qsolar3 = measurement3{(1:2160),4};
Tamb3   = measurement3{(1:2160),5};
TK1_3    = measurement3{(1:2160),6};
TK3     = measurement3{(1:2159),6};
Phi3    = measurement3{(1:2160),7};

%data for Q4
observations=readtable('measurements.csv');

q_dot_occ=table2array(observations([1:end-1],1));
q_dot_ac=table2array(observations([1:end-1],2));
q_dot_vent=table2array(observations([1:end-1],3));
q_dot_solar=table2array(observations([1:end-1],4));
T_amb=table2array(observations([1:end-1],5));
T_b=table2array(observations(:,6));


%% Q1_a & Q1_b

disp("Q1_a &Q1_b:")
c_1b = [-4 -2.5];
A_1b = [1 1;3000 1500];
b_1b = [12 24000 + 300*E1]';
lb_1b = [0 0]';
ub_1b = [inf inf]';
[x_1b,val_1b,flag_1b] = ...
    linprog(c_1b, A_1b, b_1b, [], [], lb_1b, ub_1b, [], line_optim_opt);
possi_x_1b=[5,6,6;7,5,6];
possi_power=-c_1b*possi_x_1b;
cost=A_1b(2,:)*possi_x_1b;

disp("without considering integral")
disp("installation plan");
disp(x_1b);
```

```
    disp("maximum power");
    disp(-val_1b);

    disp("possible integer installation plan:");
    disp(possi_x_1b);
    disp("possible power:");
    disp(possi_power);
    disp("they cost:");
    disp(cost);
    disp("the third plan is not feasible, so ")

    disp("installation plan");
    disp(possi_x_1b(:,1));
    disp("maximum power");
    disp(possi_power(1));

    disp("%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%");

    %% Q1_c
    disp("Q1_c");
    maintenance=eval(maintenance);

    acc_maintenance=cumsum(maintenance,2); %%maintenance accumulation by ...
        years
    budget=(4000+100*E1)*ones(1,10);
    acc_budget=cumsum(budget); %%maintenance budget accumulation by years

    x_1c=[];
    val_1c=[];
    flag_1c=[];
    f_1c=-[4,2.5];

    for i=1:1:10
        A_1c=[1,1;3000+acc_maintenance(1,i),1500+acc_maintenance(2,i)];
        b_1c=[12,24000+300*E1+acc_budget(1,i)];
        lb_1c=[0,0]';
        ub_1c=[];
        [tem_x,tem_val,tem_flag]=...
            linprog(f_1c,A_1c,b_1c,[],[],lb_1c,ub_1c,line_optim_opt);
        x_1c=[x_1c,tem_x];
        val_1c=[val_1c,tem_val];
        flag_1c=[flag_1c,tem_flag];
    end

    disp('installation plan for different duration:');
    disp(x_1c);
    disp('maximum power for different duration');
    disp(-val_1c);
    disp('optimization flag for different duration');
    disp(flag_1c);

    duration=max(-val_1c);

    disp("without considering the answer should be integral:")
    [opt_val,duration_time]=max(-val_1c);
    max_power_1c=x_1c(:,duration_time);
    disp("the best durable years");
    disp(duration_time);
    disp("best installation plan");
    disp(max_power_1c);
```

```
    disp('maximum power');
    disp(max(-val_1c))

    %integral answer for duration time 5
    possi_instal_plans=[11,11,12;0,1,0]; %possible integer solutions for ...
        durable years 5
    A_cost_1c=[3000+acc_maintenance(1,5),1500+acc_maintenance(2,5)];
    b_cost_1c=24000+300*E1+acc_budget(1,5);
    possi_cost=zeros(1,3);
    possi_power=zeros(1,3);

    for i=1:1:3
        possi_cost(i)=A_cost_1c*possi_instal_plans(:,i);
        possi_power(i)=[4,2.5]*possi_instal_plans(:,i);
        if possi_cost(i)>b_cost_1c %whether meet constraints or not
            possi_cost(i)=-1;
            possi_power(i)=-1;
        end
    end

    [max_power_1c,pos]=max(possi_power);
    instal_plan_1c=possi_instal_plans(:,pos);

    disp("considering integer constraints, for 5 years:");
    disp("optimal integer installation plan:");
    disp(instal_plan_1c);
    disp("optimal power:");
    disp(max_power_1c);
    disp("However, if duration time is chosen to be 4,6"+...
        " it seems also possible to obtain this plan");

    %test other duration years
    possi_cost=zeros(3,3);
    possi_power=zeros(1,3);
    for i=1:1:3
        A_cost_1c=[3000+acc_maintenance(1,i+3),1500+acc_maintenance(2,i+3)];
        b_cost_1c=24000+300*E1+acc_budget(1,i+3);
        for j=1:1:3
            possi_cost(i,j)=A_cost_1c*possi_instal_plans(:,j);
            possi_power(i,j)=[4,2.5]*possi_instal_plans(:,j);
            if possi_cost(i,j)>b_cost_1c
                possi_cost(i,j)=-1;
                possi_power(i,j)=-1;
            end
        end
    end

    disp("check all possible install plan for duration time 4,5,6");
    disp(possi_instal_plans);
    disp("check all possible maximum power for duration time 4,5,6");
    disp(possi_power);
    disp("And now we can find, for duration time 4,5,6,7, choose 11 x_1b ...
        and 1 Y "+...
        "we can obtain optimal power 46.5kW");

    disp("%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%");

    %% Q3
    disp("Q3:");
    %task3
```

```
    Qsigma3 = Qocc3(1:2159,1) - Qvent3(1:2159,1) + Qac3(1:2159,1);
    Tsigma3 = Tamb3(1:2159,1) - TK3(1:2159,1);

    phi_3 = Δ_t*[Qsolar3(1:2159,1) Qsigma3 Tsigma3];
    Y_3 = TK1_3(2:2160,1) - TK3(1:2159,1);
    H_3 = 2*(phi_3') * phi_3;
    C_3 = -2*(Y_3') * phi_3;
    % Five = [TK Q];
    % H2 = Five' * Five;
    % c2 = -2 * TK1' * Five;
    % lb_3 = [-Inf -Inf];
    % ub_3 = [Inf Inf];
    [x_3,val_3,flag_3] =...
        quadprog(H_3, C_3, [], [], [], [], [], [], [], quad_optim_opt);

    if flag_3==1
        disp("value of a1,a2,a3 are:");
        disp(x_3);
    else
        disp("no optimal solutions");
    end
    disp("%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%");

    %% Q_4
    disp("Q4")

    N=2160;
    T1=22.43;
    additional_cost=0.1+13/10;
    price_phi=table2array(observations(:,end));

    T_min=ones(N-1,1)*15;
    T_max=ones(N-1,1)*28;
    T_ref=ones(N-1,1)*22;
    q_dot_ac_max=ones(N,1)*100;
    lb_4=[zeros(N,1);T_min];
    ub_4=[q_dot_ac_max;T_max];

    for i=1:1:N-1
        if (q_dot_occ(i)<0)
            lb_4(N+i-1)=-inf;
            ub_4(N+i-1)=inf;
        end
    end

    c_4=[price_phi;-2*additional_cost*T_ref];

    H_4=zeros(2160*2-1,2160*2-1);
    for i=2161:2*2160-1
        H_4(i,i)=additional_cost*2;
    end

    parameter_A=1-x_3(3)*Δ_t;

    parameter_B=[x_3(1)*Δ_t,x_3(2)*Δ_t,x_3(2)*Δ_t,...
        -x_3(2)*Δ_t,x_3(3)*Δ_t];

    beq_4=parameter_B(1)*q_dot_solar+parameter_B(2)*q_dot_occ+...
        parameter_B(4)*q_dot_vent+parameter_B(5)*T_amb;
    beq_4(1)=beq_4(1)+T1*parameter_A;
```

```matlab

    A1=diag(-parameter_B(3)*ones(1,N-1));
    A2=zeros(N-1,1);
    A3=diag(ones(1,N-1))+diag(-ones(1,N-1-1)*parameter_A,-1);
    Aeq_4=[A1,A2,A3];

    [x_4,fval_4,flag_4]=...
        quadprog(H_4,c_4,[],[],Aeq_4,beq_4,lb_4,ub_4,[],quad_optim_opt);

    if flag_4==1
        fval_4=fval_4+1.4*2160*22^2+1.4*T1^2-2.8*22*T1;
        disp("the optimal cost for air-conditioning along the horizon of N steps");
        disp(fval_4);
    else
        disp("no optimal solutions");
    end
```