SC42056: Linear and Quadratic Programming Assignment

Alexandra Ministeru, Weihong Tang

October 3, 2021

This assignment represents an application of Linear and Quadratic programming techniques to a problem related to the cryptocurrency field. The last three digits of the student numbers are 432 and 160 respectively, such that E_1, E_2 and E_3 parameters have the following values:

$$E_1 = 5, E_2 = 9, E_3 = 2$$

1 Assignment Solution

1.1 Configuring the mining setup

The maximum budget and the available cluster configuration options introduce constraints that will have to be considered further on. Using the value of E_1, the budget is evaluated to have a maximum value of 2100€.

1.2 Fastest mining rate

The GreenMine has a mining capacity of 3.5 coins per day, while that of the RedMine is 2.2 coins per day. The total mining rate depends on the number of mining devices of each type used. By denoting the amount of GreenMines with G and RedMines with R, the mining rate can be expressed as in 1.

$$miningRate = 3.5G + 2.2R \tag{1}$$

Finding the fastest mining rate can be achieved by maximizing equation 1 with respect to G and R. The total number of mining devices is limited to 10, while the maximum budget is 2100€. Additionally, G and R should have positive values. This results in the optimization problem illustrated in 2.

$$\begin{gathered}
\max_{G,R} 3.5G + 2.2R \\
G + R \leq 10 \\
250G + 120R \leq 2100 \\
G, R \geq 0
\end{gathered} \tag{2}$$

The objective function and the constraints depend linearly on G and R. Therefore, they can be formulated as a linear programming problem. Prior to solving, rewriting the problem into standard form is necessary. Therefore, the maximization is transformed into a minimization and for each inequality a dummy variable is introduced such that constraints are represented by equalities, as shown in 3.

$$\begin{gathered}
\min_{G,R} -3.5G - 2.2R \\
G + R + s_1 = 10 \\
250G + 120R + s_2 = 2100 \\
G, R, s_1, s_2 \geq 0
\end{gathered} \tag{3}$$

1.3 Optimal solution

Solving the optimization problem using MATLAB requires writing the objective function and the constraints into matrix form. Rearranging relations 3 using 4 results in the A, b, and c matrices 5. The four variables to be obtained are inferiorly bounded by zero.

$$\begin{aligned} &\min_x c^T x \\ &Ax = b \\ &x \geq 0 \end{aligned} \tag{4}$$

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 250 & 120 & 0 & 1 \end{bmatrix}, b = \begin{bmatrix} 10 & 2100 \end{bmatrix}, c = \begin{bmatrix} -3.5 & -2.2 & 0 & 0 \end{bmatrix} \tag{5}$$

The values G and R are required to be integers, since they correspond to a number of mining devices. Nevertheless, using the MATLAB function *linprog* leads to a real solution:

$$G = 6.9231,\ R = 3.0769,\ miningRate = 31 \text{ coins/day}.$$

The optimum point corresponds to a vertex of the feasible set [vdBS21]. Therefore, for certain problems it may be possible to approximate an integer solution based on the real one such that the objective function value remains in the vicinity of the optimum point. However, this approach may lead to incorrect solutions depending on the functions that determine the feasible set. In some cases, the nearest integer solution contained in the feasible set might not be close enough to the real optimum point. In consequence, specific integer linear programming techniques should be employed. In this particular case, the approximation was completed by rounding the real values to near integers and checking whether the constraints are still fulfilled, leading to the following solution:

$$G = 6,\ R = 4,\ miningRate = 26.8 \text{ coins/day}.$$

The setup is limited by both the maximum budget and the maximum amount of mining devices. The mining rate would increase by using only GreenMiners, which is not possible giving the constraint on the budget. Eliminating the budget limitation leads to a MATLAB solution of 10 GreenMiners and zero RedMiners. Conversely, eliminating the superior limit on the device number results in a solution that consists of 18 RedMiners (because they are the cheapest) and zero GreenMiners.

1.4 Balanced mining

The one-year tuition fee amounts to 8180€. Balancing the system to mine the required amount over the course of 365 days is equivalent to earning approximately 22.411 coins per day, which results in a constraint regarding the mining rate. Obtaining the cheapest setup can be achieved by minimizing the cost of the mining devices. The new linear programming problem can be therefore formulated as in 6.

$$\begin{aligned} &\min_{G,R} 250G + 120R \\ &3.5G + 2.2R = \frac{8180}{365} \\ &G + R + s_1 = 10 \\ &G, R, s_1 \geq 0 \end{aligned} \tag{6}$$

The solution identified in MATLAB using *linprog* is, once again, real:

$$G = 0.3161,\ R = 9.6839,\ cost = 1241\,€.$$

Approximating the number of mining devices to integers causes an error between the intended mining rate per day and the actual resulting rate. Rounding G to 1 and R to 9 results in 0.889 less DelftCoins per day, which would lead to not fulfilling the yearly fee. Alternatively, approximating G to 0 and R to 10 leads to a surplus of 0.411 DelftCoins per day, with a total cost of 1200€. The latter is evaluated to be the appropriate solution, since it allows achieving the goal of paying the fee over one year using cryptocurrency.

$$G = 0,\ R = 10,\ cost = 1200\,€.$$

2 Obtaining a discrete-time model for a single mining device

The continuous-time dynamic model is given by the differential equation represented in 7,

$$\frac{\mathrm{d}T(t)}{\mathrm{d}t} = a_1 \left[T^{\mathrm{amb}}(t) - T(t)\right] + a_2 \left[\dot{q}^{\mathrm{in}}(t) - \dot{q}^{\mathrm{out}}(t)\right] \tag{7}$$

where a_1 and a_2 are model parameters and $\dot{q}^{\mathrm{out}}(t), \dot{q}^{\mathrm{in}}(t) > 0$.

Transforming the continuous-time dynamic model into a discrete-time version can be done by using forward difference, as illustrated in 8.

$$\frac{\mathrm{d}T_k}{\mathrm{d}t} \approx \frac{\mathrm{d}T_{k+1} - \mathrm{d}T_k}{\Delta t} = a_1 T_k^{\mathrm{amb}} - a_1 T_k + a_2 \dot{q}_k^{\mathrm{in}} - a_2 \dot{q}_k^{\mathrm{out}} \tag{8}$$

Next, relation 9 can be obtained.

$$T_{k+1} = (-\Delta t a_1 + 1)T_k + \Delta t(a_2 \dot{q}_k^{\mathrm{in}} - a_2 \dot{q}_k^{\mathrm{out}} + T_k^{\mathrm{amb}}) \tag{9}$$

Therefore, the resulting model has to be of the form:

$$T_{k+1} = AT_k + B\left[\dot{q}_k^{\mathrm{in}}, \dot{q}_k^{\mathrm{out}}, T_k^{\mathrm{amb}}\right]^\top, \tag{10}$$

where $A = -\Delta t a_1 + 1$ and $B = \Delta t\,[a_2,\ -a_2,\ a_1]$.

3 Identifying model parameters of a single mining device

If $[\dot{q}_k^{\mathrm{in}}, \dot{q}_k^{\mathrm{out}}, T_k^{\mathrm{amb}}]^T$ is denoted by u_k, the optimization problem can be rewritten as in 11 .

$$\min_{a_1,a_2} \sum_{k=1}^{N} (T_{k+1} - (AT_k + Bu_k))^2 \tag{11}$$

Expanding expression 11 leads to 12.

$$\min_{a_1,a_2} \sum_{k=1}^{N} T_{k+1}^2 - 2T_{k+1}(AT_k + Bu_k) + (AT_k)^2 + 2AT_k Bu_k + (Bu_k)^2 \tag{12}$$

The elements of matrices A and B consist exclusively of terms containing a_1 and a_2. Therefore, expression 12 can be formulated as a quadratic programming problem with no constraints. Given that 13 is the standard form of a quadratic programming problem, the quadratic part can be identified as $\sum_{k=1}^{N} (AT_k)^2 + 2AT_k Bu_k + (Bu_k)^2$, while the linear part is $\sum_{k=1}^{N} -2T_{k+1}(AT_k + Bu_k)$.

$$\min_x \frac{1}{2} x^T H x + c^T x \tag{13}$$

Starting from the quadratic part, the Hessian matrix $H \in R^{4x4}$ is constructed based on the coefficients of the terms A^2, AB and B^2. Vector c consists of the linear part coefficients. The following notations are used for the internal temperature vector and for the input temperature matrix:

$$T_{vk} = [T_1, T_2, ..., T_N]^T$$
$$T_{vk'} = [T_2, T_3, ..., T_{N+1}]^T$$
$$U_m = \begin{bmatrix} \dot{q}_k^{\mathrm{in}} & \dot{q}_k^{\mathrm{out}} & T_k^{\mathrm{amb}} \\ \vdots & \vdots & \vdots \\ \dot{q}_k^{\mathrm{in}} & \dot{q}_k^{\mathrm{out}} & T_k^{\mathrm{amb}} \end{bmatrix}$$

Therefore, matrix H and vector c can be expressed as in relation 14.

$$H = \begin{bmatrix} 2T_{vk}^T T_{vk} & 2T_{vk}^T U_m \\ 2U_m^T T_{vk} & 2U_m^T U_m \end{bmatrix},\ c = \begin{bmatrix} -2T_{vk'}^T T_{vk} & -2T_{vk'}^T U_m \end{bmatrix} \tag{14}$$

The term T_{k+1}^2 can be disregarded while solving the quadratic programming problem, since it is constant and does not influence the final solution. Employing the MATLAB function *quadprog* leads to the solution 15, which is equivalent to $[A\ B^T](\Delta t = 60s)$.

$$x = \begin{bmatrix} 0.9745 \\ 0.1250 \\ -0.1247 \\ 0.0245 \end{bmatrix} \tag{15}$$

Consequently, the estimated values of the parameters are $a_1 = 4.088e-04$ and $a_2 = 0.0021$.

4 Optimizing the cost for mining of a single mining device

Maximizing the mining rate and minimizing the miner operating cost can be done by solving the optimization problem and the constraints given by 16.

$$\min_{\substack{T_2,\ldots,T_{N+1} \\ \dot{q}_1^{in},\ldots,\dot{q}_N^{in}}} \sum_{k=1}^{N} \Phi_k \left(\dot{q}_k^{\text{in}} + \dot{q}_k^{\text{out}}\right) \Delta t - \Psi \dot{q}_k^{\text{in}} \Delta t \tag{16}$$

$$\begin{aligned} \text{s.t.} \quad & T_{k+1} = AT_k + B\left[\dot{q}_k^{\text{in}}, \dot{q}_k^{\text{out}}, T_k^{\text{amb}}\right]^\top, && k = 1, \ldots, N \\ & 0 \le \dot{q}_k^{\text{in}} \le \dot{q}_{\max} = 125, && k = 1, \ldots, N \\ & T_k \le T^{\max} = 90, && k = 2, \ldots, N+1 \\ & T_1 = 23.3, N = 1440 \end{aligned}$$

The term $\Phi_k \dot{q}_k^{\text{out}} \Delta t$ is constant. Therefore, the objective function can be simplified to the following form:

$$\min_{\substack{T_2,\ldots,T_{N+1} \\ \dot{q}_1^{in},\ldots,\dot{q}_N^{in}}} \sum_{k=1}^{N} (\Phi_k - \Psi) \dot{q}_k^{\text{in}} \Delta t \tag{17}$$

The objective function along with the equation of the mining device model and the constraints depend linearly on the terms $\dot{q}_k^{in}$ and T_k. Thus, this can be regarded as a linear programming problem.

The following relations can be deducted according to $T_{k+1} = AT_k + B\left[\dot{q}_k^{\text{in}}, \dot{q}_k^{\text{out}}, T_k^{\text{amb}}\right]^\top, k = 1, \ldots, N$:

$$\begin{aligned} T_2 &= AT_1 + B_1 \dot{q}_1^{\text{in}} + B_{23}\left[\dot{q}_1^{\text{out}}, T_1^{\text{amb}}\right]^\top, \\ T_3 &= AT_2 + B_1 \dot{q}_2^{\text{in}} + B_{23}\left[\dot{q}_2^{\text{out}}, T_2^{\text{amb}}\right]^\top, \\ &\ldots \\ T_N &= AT_{N-1} + B_1 \dot{q}_{N-1}^{\text{in}} + B_{23}\left[\dot{q}_{N-1}^{\text{out}}, T_{N-1}^{\text{amb}}\right]^\top, \\ T_{N+1} &= AT_N + B_1 \dot{q}_N^{\text{in}} + B_{23}\left[\dot{q}_N^{\text{out}}, T_N^{\text{amb}}\right]^\top \end{aligned}$$

Subsequently, the unknown variables in the above equations are placed on the left hand side of the equality, whereas the known variables are on the right hand side, resulting in the following relations:

$$\begin{aligned} B_1 \dot{q}_1^{\text{in}} - T_2 &= -AT_1 - B_{23}\left[\dot{q}_1^{\text{out}}, T_1^{\text{amb}}\right]^\top, \\ B_1 \dot{q}_2^{\text{in}} + AT_2 - T_3 &= -B_{23}\left[\dot{q}_2^{\text{out}}, T_2^{\text{amb}}\right]^\top, \\ &\ldots \\ B_1 \dot{q}_{N-1}^{\text{in}} + AT_{N-1} - T_N &= -B_{23}\left[\dot{q}_{N-1}^{\text{out}}, T_{N-1}^{\text{amb}}\right]^\top, \\ B_1 \dot{q}_N^{\text{in}} + AT_N - T_{N+1} &= -B_{23}\left[\dot{q}_N^{\text{out}}, T_N^{\text{amb}}\right]^\top \end{aligned} \tag{18}$$

The MATLAB implementation of the solution uses the linear programming function *linprog(f,A,b,Aeq,beq, lb,ub)*. The coefficient vector c is illustrated in 19 and consists of two parts. The first part contains coefficients of $\dot{q}_k^{in}$ as they appear in the objective function, while the second part is comprised of zeros corresponding to temperatures.

$$f = [\Phi_1 - \Psi, \Phi_2 - \Psi, \cdots, \Phi_{1440} - \Psi, \underbrace{0, 0, \ldots, 0}_{1440}] \tag{19}$$

The linear equality constraints are expressed using matrices A_{eq} and b_{eq} which contain the coefficients in relations given by 18.

$$A_{eq} = \left[\underbrace{\begin{matrix} B_1 & 0 & \ldots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \ldots & 0 & B_1 \end{matrix}}_{1440} \quad \underbrace{\begin{matrix} -1 & 0 & \ldots & \ldots & 0 \\ A & \ddots & & & \vdots \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & A & -1 \end{matrix}}_{1440}\right]$$

$$b_{eq} = \begin{bmatrix} -AT_1 - B_{23}\left[\dot{q}_1^{\text{out}}, T_1^{\text{amb}}\right]^\top \\ -B_{23}\left[\dot{q}_2^{\text{out}}, T_2^{\text{amb}}\right]^\top \\ \vdots \\ -B_{23}\left[\dot{q}_N^{\text{out}}, T_N^{\text{amb}}\right]^\top \end{bmatrix}$$

The lower bounds are defined by vector lb. Since an inferior limit for temperature was not specified, lb only contains 1440 elements corresponding to variables $\dot{q_k^{in}}$. The rest of the vector will be completed automatically with $-Inf$ values.

$$lb = [\underbrace{0, 0, \cdots, 0}_{1440}]$$

Vector ub defines the upper bounds as specified by the power consumption and temperature constraints.

$$ub = [\underbrace{125, 125, \cdots, 125}_{1440}, \underbrace{90, 90, \ldots 90}_{1440}]$$

By employing the MATLAB function *quadprog*, the value of the simplified objective function is -0.2784 (the value of the original objective function value is 0.1151). The optimized total cost of energy is 0.8686€, obtained using 20.

$$\sum_{k=1}^{N} \Phi_k(\dot{q_k^{in}} + \dot{q_k^{out}})\Delta t \tag{20}$$

References

[vdBS21] Ton van den Boom and Bart De Schutter. Lecture Notes for the Course SC42056, September 2021.