EOAL final project

by Dmitry Beresnev, d.beresnev@innopolis.university

by Vsevolod Klyushev, v.klyushev@innopolis.university

Introduction
Problem Formulation
Theoretical Part: theoretical analysis of the given (formal) problem
    Definition of cost function
    Proof of convexity of set of admissible solutions
    Functional equation
    Proof the convexity of functional
    Possible theoretical solution: KKT theorem
Numerical Part: numerical analysis of the formal problem using the facts obtained in the previous section
Computational Studies
Conclusion
References

Introduction

We decided to work on the practical task “Optimal warehousing (logistics)” using the methods of non-linear optimization theory (NO).

The description of our task is the following:

A company sells a product. The selling process is as follows

- selling is determined for discrete time-instants: $t_0 < t_1 < ... < t_N$ and the warehouse is open for a supply at the beginning of every operating interval $[t_i, t_{i+1}]$
- the purpose is: to optimize the total costs of warehousing, the warehousing costs for every time-interval are given by $f_i(z_i, u_i)$
- z_i is the stock of the product at t_i before the new-supply, r_i is the demand of the product in $[t_i, t_{i+1}]$, u_i is the delivered quantity at t_i, $z_0 = a \geq 0$ is the initial quantity of the product

We have to formalize the given qualitative problem in the form of a NO problem, apply a suitable numerical methods (NM) and present a numerical solution.

Problem Formulation

The function $f_i(z_i, u_i)$ is defined on the N time intervals: $[t_0, t_1], [t_1, t_2], \ldots, [t_{N-1}, t_N]$. We assume, that $f_0(z_o, u_0)$ corresponds to time interval $[t_0, t_1]$, and so on. Then $f_{N-1}(z_{N-1}, u_{N-1})$ computes the cost for the time interval $[t_{N-1}, t_N]$.

We have the following restrictions: the stock products before new supply should not be negative. What means the following

$$\begin{cases} z_0 \geq 0 \\ z_1 \geq 0 \\ \ldots \\ z_N \geq 0 \end{cases}$$

At the same time. the stock of products at t_i can be defined as the difference of supply and demand at t_{i-1} plus all remaining products before t_i.

$$z_i = \begin{cases} z_{i-1} + u_{i-1} - r_{i-1},\ if\ i > 0 \\ \quad a,\ where\ a \geq 0,\ if\ i = 0 \end{cases} \tag{1}$$

Note that we assume that $r \in \mathbb{R}_+^N$ is given.

Now we can represent the constraints as following:

$$\begin{cases} z_0 \geq 0 \\ z_1 \geq 0 \\ \dots \\ z_N \geq 0 \end{cases} \Leftrightarrow \begin{cases} a \geq 0 \\ z_0 + u_0 - r_0 \geq 0 \\ \dots \\ z_{N-1} + u_{N-1} - r_{N-1} \geq 0 \end{cases}$$

By the problem description, $z_0 = a \geq 0$, so we can exclude it from the consideration. Using the equation (1), we can represent z as following

$$\begin{cases} z_1 = z_0 + u_0 - r_0 = a + u_0 - r_0 \\ z_2 = z_1 + u_1 - r_1 = a + u_0 + u_1 - r_0 - r_1 \\ z_3 = z_2 + u_2 - r_2 = a + \sum_{i=0}^{2} u_i - \sum_{i=0}^{2} r_i \\ \dots \\ z_k = a + \sum_{i=0}^{k-1} u_i - \sum_{i=0}^{k-1} r_i \\ \dots \\ z_N = a + \sum_{i=0}^{N-1} u_i - \sum_{i=0}^{N-1} r_i \end{cases}$$

So we can rewrite our constraints as follows

$$z_k = a + \sum_{i=0}^{k-1} u_i - \sum_{i=0}^{k-1} r_i \geq 0, k = 1, \dots, N \tag{2}$$

Considering all the observations made before, we can formalize the optimization problem in the following way:

$$\text{Minimize } J(u) = \sum_{i=0}^{N-1} f_i(z_i, u_i) \tag{3}$$

$$\text{subject to } u \in \mathscr{A} = \{u \in \mathbb{R}_+^N \mid g(u) \leq 0\},$$

$$\text{where } g(u) = (-a - \sum_{i=0}^{k-1} u_i + \sum_{i=0}^{k-1} r_i), k = 1, \dots, N \tag{4}$$

Theoretical Part: theoretical analysis of the given (formal) problem

Definition of cost function

Let us introduce the following cost function

$$f_i(z_i, u_i) = \alpha_i \cdot z_i + \beta_i \cdot u_i^2 + \gamma_i$$

where $\alpha_i, \beta_i, \gamma_i$ are parameters, such that $\alpha, \beta, \gamma \in \mathbb{R}_+^{N-1}$.

The motivation for using such class of functions is the following: the relevant cost function should have some weights for the new supply products (α) and for the products which are already in the warehouse (β), and also some constant cost (γ). Moreover, we use u_i^2 to make our function non-linear. Also, as the time intervals are different, it is reasonable to assume that the parameters α, β, γ should also depend on these intervals. The simplest case may be, for example, when the longer time interval is the higher the cost is and the bigger the parameters are. Therefore, we assume that parameters

$\alpha_i, \beta_i, \gamma_i$ for all $i = 0, 1, \ldots, N-1$ are fully defined by the time interval $[t_i, t_{i+1}]$, in other words $\alpha_i = \alpha_i(t_i, t_{i+1}),\ \beta_i = \beta_i(t_i, t_{i+1}),\ \gamma_i = \gamma_i(t_i, t_{i+1})$ for all $i = 0, 1, \ldots, N-1$. Since all the time intervals are given, we assume $\alpha_i, \beta_i, \gamma_i,$ for all $i = 0, 1, \ldots, N-1$ to be constant, and the vectors

$$\alpha = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \ldots \\ \alpha_{N-1} \end{bmatrix}, \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \ldots \\ \beta_{N-1} \end{bmatrix}, \gamma = \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \ldots \\ \gamma_{N-1} \end{bmatrix}$$

are constants too.
Note also that since f_i determines the cost for the interval $[t_i, t_{i+1}]$, the function f_N is undefined, as it corresponds to the interval $[t_N, +\infty)$ and can not be practically evaluated using the assumptions made above.

Let us now use (2) to make a cost function to depend only on components of u.

$$f_i(z_i, u_i) = \alpha_i \cdot z_i + \beta_i \cdot u_i^2 + \gamma_i$$

$$f_i(u_0, u_1, \ldots, u_i) = \begin{cases} \alpha_i \cdot (a + \sum_{j=0}^{i-1} u_j - \sum_{j=0}^{i-1} r_j) + \beta_i \cdot u_i^2 + \gamma_i,\ i = 1, 2, \ldots, N-1 \\ \alpha_0 \cdot a + \beta_0 \cdot u_0^2 + \gamma_0,\ i = 0 \end{cases}$$

$$f_i(u_0, u_1, \ldots, u_i) = \begin{cases} \alpha_i a + \alpha_i \sum_{j=0}^{i-1} u_j - \alpha_i \sum_{j=0}^{i-1} r_j + \beta_i \cdot u_i^2 + \gamma_i,\ i = 1, 2, \ldots, N-1 \\ \alpha_0 \cdot a + \beta_0 \cdot u_0^2 + \gamma_0,\ i = 0 \end{cases} \tag{5}$$

Proof of convexity of set of admissible solutions

Let's prove that the set $\mathscr{A} \subset \mathbb{R}^N$ is convex using the definition: The set $\mathscr{A} \subset \mathbb{R}^N$ is called convex if for all $x_1, x_2 \in \mathscr{A}$

$$x_3 := \lambda x_1 + (1 - \lambda) x_2 \in \mathscr{A}$$
$$\forall \lambda \in [0, 1]$$

Let us take $x_1, x_2 \in \mathscr{A}$. Ten let us consider $x_3 = \lambda x_1 + (1 - \lambda) x_2$, where $\lambda \in [0, 1]$. Recall that $\mathscr{A} = \{u \in \mathbb{R}_+^N \mid g(u) \leq 0\}$

If $x_1, x_2 \in \mathbb{R}_+^N$, it is obvious that their linear combination with non-negative coefficients x_3 has also lie in $\mathbb{R}_+^N$. Therefore, if we can prove that $g(x_3) \leq 0$ holds, then $x_3 \in \mathscr{A}$ and $\mathscr{A}$ is a convex set.

So let us prove it.

Recall, that $g(u)$ is defined as

$$g(u) = (-a - \sum_{i=0}^{k-1} u_i + \sum_{i=0}^{k-1} r_i) \leq 0, \text{ for all } k = 1, 2, \ldots N$$

Then we need to prove the following

$$g(x_3) = (-a - \sum_{i=0}^{k-1} x_{3_i} + \sum_{i=0}^{k-1} r_i) \leq 0, \text{ for all } k = 1, 2, \ldots N$$

where x_{3_i} is an i_{th} component of the vector x_3. Obviously, that

$$x_3 = \lambda x_1 + (1 - \lambda) x_2 \Leftrightarrow x_{3_k} = \lambda x_{1_k} + (1 - \lambda) x_{2_k}, \text{ for all } k = 1, 2, \ldots N$$

Therefore, we can prove the general case for $\forall k = 1, 2, \ldots N$, and all the concrete cases will automatically hold:

$$-a - \sum_{i=0}^{k-1} x_{3_i} + \sum_{i=0}^{k-1} r_i \leq 0$$

$$-a - \sum_{i=0}^{k-1}(\lambda x_{1_i} + (1-\lambda)x_{2_i}) + \sum_{i=0}^{k-1} r_i \leq 0$$

$$-a - \lambda\sum_{i=0}^{k-1} x_{1_i} - (1-\lambda)\sum_{i=0}^{k-1} x_{2_i} + \sum_{i=0}^{k-1} r_i \leq 0$$

$$-\lambda a - (1-\lambda)a - \lambda\sum_{i=0}^{k-1} x_{1_i} - (1-\lambda)\sum_{i=0}^{k-1} x_{2_i} + \lambda\sum_{i=0}^{k-1} r_i + (1-\lambda)\sum_{i=0}^{k-1} r_i \leq 0$$

$$\lambda(-a - \sum_{i=0}^{k-1} x_{1_i} - \sum_{i=0}^{k-1} r_i) + (1-\lambda)(-a - \sum_{i=0}^{k-1} x_{2_i} - \sum_{i=0}^{k-1} r_i) \leq 0$$

$$\lambda g(x_1) + (1-\lambda)g(x_2) \leq 0 \tag{6}$$

Since $x_1, x_2 \in \mathscr{A}$ conditions $g(x_1) \leq 0$ and $g(x_2) \leq 0$ holds, and λ and $(1-\lambda)$ are both non-negative, equation (6) also holds. Therefore, $\mathscr{A}$ is a convex set.

Functional equation

Using the (5), we get

$$J(u) = \sum_{i=0}^{N-1} f_i(u_0, \ldots, u_i)$$

$$J(u) = \alpha_0 \cdot a + \beta_0 \cdot u_0^2 + \gamma_0 + \sum_{i=1}^{N-1}(\alpha_i \cdot (a + \sum_{j=0}^{i-1} u_j - \sum_{j=0}^{i-1} r_j) + \beta_i \cdot u_i^2 + \gamma_i)$$

$$J(u) = a\sum_{i=0}^{N-1}\alpha_i + \sum_{i=0}^{N-1}\gamma_i + \sum_{i=0}^{N-1}\beta_i u_i^2 + \sum_{i=1}^{N-1}(\alpha_i \sum_{j=0}^{i-1}(u_j - r_j))$$

We can rewrite the last term as following:

$$\begin{gathered}
\sum_{i=1}^{N-1}(\alpha_i \sum_{j=0}^{i-1}(u_j - r_j)) = \alpha_1(u_0 - r_0) + \\
\alpha_2(u_0 - r_0) + \alpha_2(u_1 - r_1) + \\
\alpha_3(u_0 - r_0) + \alpha_3(u_1 - r_1) + \alpha_3(u_2 - r_2) + \\
\cdots + \\
\alpha_{N-1}(u_0 - r_0) + \cdots + \alpha_{N-1}(u_{N-2} - r_{N-2}) \\
= \sum_{i=0}^{N-2}((u_i - r_i)\sum_{j=i+1}^{N-1}\alpha_j) \\
= \sum_{i=0}^{N-2}(u_i \sum_{j=i+1}^{N-1}\alpha_j) - \sum_{i=0}^{N-2}(r_i \sum_{j=i+1}^{N-1}\alpha_j)
\end{gathered}$$

So we can write the equation for the objective functional

$$J(u) = a\sum_{i=0}^{N-1}\alpha_i + \sum_{i=0}^{N-1}\gamma_i + \sum_{i=0}^{N-1}\beta_i u_i^2 + \sum_{i=0}^{N-2}(u_i \sum_{j=i+1}^{N-1}\alpha_j) - \sum_{i=0}^{N-2}(r_i \sum_{j=i+1}^{N-1}\alpha_j) \tag{7}$$

We can use the vectors and matrices to simplify the equation:

$$J(u) = a \cdot e_N^T\alpha + e_N^T\gamma + \beta^T\hat{u} + u^T\hat{\alpha} - r^T\hat{\alpha} \tag{8}$$

where $\hat{\alpha} = \begin{bmatrix} \sum_{i=1}^{N-1} \alpha_i \\ \sum_{i=2}^{N-1} \alpha_i \\ \dots \\ \sum_{i=N-1}^{N-1} \alpha_i \\ 0 \end{bmatrix}$, $e_N = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \dots \\ 1 \end{bmatrix}$ and $dim(e_N) = N \times 1$, $\hat{u} = \begin{bmatrix} u_0^2 \\ u_1^2 \\ u_2^2 \\ \dots \\ u_{N-1}^2 \end{bmatrix}$

Proof the convexity of functional

Let's prove that for convex set $\mathscr{A}$ the functional $J : \mathscr{A} \to \mathbb{R}$ is convex using the definition:

Let $\mathscr{A}$ be a convex set, a functional $J : \mathscr{A} \to \mathbb{R}$ is called convex if for all $x_1, x_2 \in \mathscr{A}$

$$J(x_3) \leq \lambda J(x_1) + (1 - \lambda) J(x_2)$$
$$\forall \lambda \in [0, 1]$$

where $x_3 := \lambda x_1 + (1 - \lambda) x_2$

Let us show that the following is true

$$J(x_3) \leq \lambda J(x_1) + (1 - \lambda) J(x_2)$$

Using the (8) we get

$$J(\lambda x_1 + (1 - \lambda) x_2) \leq \lambda J(x_1) + (1 - \lambda) J(x_2)$$

$$ae_N^T \alpha + e_N^T \gamma + \beta^T \hat{x}_3 + (\lambda x_1 + (1 - \lambda) x_2)^T \hat{\alpha} - r^T \hat{\alpha} \leq \lambda (ae_N^T \alpha + e_N^T \gamma + \beta^T \hat{x}_1 + x_1^T \hat{\alpha} - r^T \hat{\alpha}) + (1 - \lambda)(ae_N^T \alpha +$$

where $\hat{x}_1 = \begin{bmatrix} x_{1_0}^2 \\ x_{1_1}^2 \\ x_{1_2}^2 \\ \dots \\ x_{1_{N-1}}^2 \end{bmatrix}$, $\hat{x}_2 = \begin{bmatrix} x_{2_0}^2 \\ x_{2_1}^2 \\ x_{2_2}^2 \\ \dots \\ x_{2_{N-1}}^2 \end{bmatrix}$ and $\hat{x}_3 = \begin{bmatrix} (\lambda x_{1_0} + (1 - \lambda) x_{2_0})^2 \\ (\lambda x_{1_1} + (1 - \lambda) x_{2_1})^2 \\ (\lambda x_{1_2} + (1 - \lambda) x_{2_2})^2 \\ \dots \\ (\lambda x_{1_{N-1}} + (1 - \lambda) x_{2_{N-1}})^2 \end{bmatrix}$

Further we get

$$ae_N^T \alpha + e_N^T \gamma + \beta^T \hat{x}_3 + (\lambda x_1 + (1 - \lambda) x_2)^T \hat{\alpha} - r^T \hat{\alpha} \leq (\lambda + 1 - \lambda) ae_N^T \alpha + (\lambda + 1 - \lambda) e_N^T \gamma - (\lambda + 1 - \lambda) r^T \hat{\alpha} +$$

$$\beta^T \hat{x}_3 + (\lambda x_1 + (1 - \lambda) x_2)^T \hat{\alpha} \leq \lambda x_1^T \hat{\alpha} + (1 - \lambda) x_2^T \hat{\alpha} + \lambda \beta^T \hat{x}_1 + (1 - \lambda) \beta^T \hat{x}_1$$

$$(\lambda x_1 + (1 - \lambda) x_2)^T \hat{\alpha} + \beta^T \hat{x}_3 \leq (\lambda x_1 + (1 - \lambda) x_2)^T \hat{\alpha} + \lambda \beta^T \hat{x}_1 + (1 - \lambda) \beta^T \hat{x}_1$$

$$\beta^T \hat{x}_3 \leq \lambda \beta^T \hat{x}_1 + (1 - \lambda) \beta^T \hat{x}_1$$

As $\forall \beta_i \geq 0$, the inequality above holds when the vector $\hat{x}_3$ is componentwise less or equal to the vector $\lambda \hat{x}_1 + (1 - \lambda) \hat{x}_2$:

$$\beta^T \hat{x}_3 \leq \lambda \beta^T \hat{x}_1 + (1 - \lambda) \beta^T \hat{x}_2 \Leftrightarrow \hat{x}_{3_k} \leq \lambda \hat{x}_{1_k} + (1 - \lambda) \hat{x}_{2_k},\ k = 0, 1, \dots, N - 1$$

If we manage to prove the last inequality for arbitrary component k, all the inequality above hold and the initial proposition is true.

$$\hat{x}_{3_k} \leq \lambda \hat{x}_{1_k} + (1 - \lambda) \hat{x}_{2_k}$$

$$(\lambda x_{1_k} + (1 - \lambda) x_{2_k})^2 \leq \lambda x_{1_k}^2 + (1 - \lambda) x_{1_k}^2$$

$$\lambda^2 x_{1_k}^2 + 2\lambda(1 - \lambda) x_{1_k} x_{2_k} + (1 - \lambda)^2 x_{2_k}^2 \leq \lambda x_{1_k}^2 + (1 - \lambda) x_{1_k}^2$$

$$\lambda^2 x_{1_k}^2 + 2\lambda(1 - \lambda) x_{1_k} x_{2_k} + ((1 - \lambda)^2 - (1 - \lambda)) x_{2_k}^2 \leq \lambda x_{1_k}^2$$

$$\lambda^2 x_{1_k}^2 + 2\lambda(1 - \lambda) x_{1_k} x_{2_k} + (1 - \lambda)((1 - \lambda) - 1) x_{2_k}^2 \leq \lambda x_{1_k}^2$$

$$\lambda^2 x_{1_k}^2 + 2\lambda(1 - \lambda) x_{1_k} x_{2_k} + (1 - \lambda)(-\lambda) x_{2_k}^2 \leq \lambda x_{1_k}^2$$

$$(\lambda^2 - \lambda) x_{1_k}^2 + 2\lambda(1 - \lambda) x_{1_k} x_{2_k} + (1 - \lambda)(-\lambda) x_{2_k}^2 \leq 0$$

$$\lambda(\lambda-1)x_{1_k}^2 + 2\lambda(1-\lambda)x_{1_k}x_{2_k} + (1-\lambda)(-\lambda)x_{2_k}^2 \leq 0$$

$$\lambda(\lambda-1)(x_{1_k}^2 - x_{2_k}^2) \leq 0$$

The final inequality is true for all x_1, x_2 and $\lambda \in [0,1]$. Therefore, the initial assumption is true and the functional J is convex.

Possible theoretical solution: KKT theorem

We has already proved that $\mathscr{A}$ and J are convex, so, by definition, the given optimization problem is convex.

Now this problem can be characterized as **nonlinear convex constrained optimization problem**.

As $J(\cdot)$ and $g(\cdot)$ both are just a linear combinations of variable u and constants, it is obvious that they both are continuously differentiable.

Recall the **KKT Theorem**:

> Consider a non-linear optimization problem. Let x^{opt} be a local minimum of this problem. Then there exists a vector of Lagrange multipliers such that x^{opt} is a solution of the following (parametric) system:
>
> $$\lambda := (\lambda_1, \dots, \lambda_m)^T \geq 0$$
>
> $$\nabla_x \mathscr{L}(x, \lambda) = 0 \text{ (stationarity)},$$
> $$\lambda_i g_i(x) = 0,\ i = 1, \dots, m \text{ (complementarity)}$$
>
> where $\mathscr{L}(\cdot, \cdot)$ is the Lagrange function $\mathscr{L}(x, \lambda) := J(x) + \sum_{i=1}^m \lambda_i g_i(x)$ associated with the given problem

Moreover, as our problem is proved to be convex and $J(\cdot)$ and $g(\cdot)$ both are continuously differentiable, we can use the following theorem:

> Consider a convex optimization problem with
>
> $$\mathscr{A} := \{x \in \mathbb{R}^n \mid g(x) \leq 0\}$$
>
> and assume that $J(\cdot)$ and $g(\cdot)$ both are continuously differentiable. Then KKT Theorem is necessary and sufficient condition for an optimal solution (global minimum) $x^{opt} \in \mathscr{A}$.

Considering two theorems mentioned above, we can conclude that the optimal solution (global minimum) u^{opt} to our problem is a solution for the following system:

$$\begin{cases} \nabla_u \mathscr{L}(u, \lambda) = 0 \\ \lambda_i g_i(u) = 0,\ i = 0, \dots, N-1 \end{cases}$$

where $\lambda := (\lambda_0, \dots, \lambda_{N-1})^T \geq 0$ and $\mathscr{L}(u, \lambda) := J(u) + \sum_{i=0}^{N-1} \lambda_i g_i(u)$.

Let's apply KKT theorem then

$$\mathscr{L}(x, \lambda) := J(x) + \sum_{i=0}^{N-1} \lambda_i g_i(x)$$

$$\mathscr{L}(u, \lambda) := a\sum_{i=0}^{N-1}\alpha_i + \sum_{i=0}^{N-1}\gamma_i + \sum_{i=0}^{N-1}\beta_i u_i^2 + \sum_{i=0}^{N-2}(u_i \sum_{j=i+1}^{N-1}\alpha_j) - \sum_{i=0}^{N-2}(r_i \sum_{j=i+1}^{N-1}\alpha_j) + \sum_{i=0}^{N-1}(\lambda_i(-a - \sum_{j=0}^{i}u_j + \sum_{j=0}^{i}r_j))$$

$$\frac{\partial \mathscr{L}}{\partial u_0} = 2\beta_0 u_0 + \sum_{j=1}^{N-1}\alpha_j - \sum_{i=0}^{N-1}\lambda_i = 0$$

$$\frac{\partial \mathscr{L}}{\partial u_1} = 2\beta_1 u_1 + \sum_{j=2}^{N-1}\alpha_j - \sum_{i=1}^{N-1}\lambda_i = 0$$

$$\dots$$

$$\frac{\partial \mathscr{L}}{\partial u_k} = 2\beta_k u_k + \sum_{j=k+1}^{N-1} \alpha_j - \sum_{i=k}^{N-1} \lambda_i = 0$$

$$\dots$$

$$\frac{\partial \mathscr{L}}{\partial u_{N-2}} = 2\beta_{N-2} u_{N-2} + \sum_{j=N-1}^{N-1} \alpha_j - \sum_{i=N-2}^{N-1} \lambda_i = \beta_{N-2} + \alpha_{N-2} - \lambda_{N-2} - \lambda_{N-1} = 0$$

$$\frac{\partial \mathscr{L}}{\partial u_{N-1}} = 2\beta_{N-1} u_{N-1} - \lambda_{N-1} = 0$$

$$\lambda_i(-a - \sum_{j=0}^{i} u_j + \sum_{j=0}^{i} r_j) = 0, i = 0, 1, \dots, N-1$$

So we would have the following system of equations:

$$\begin{cases} \lambda_{N-1} = 2\beta_{N-1} u_{N-1} \\ \lambda_k = 2\beta_k u_k + \sum_{j=k+1}^{N-1} \alpha_j - \sum_{i=k+1}^{N-1} \lambda_i, k = 0, 1, \dots, N-2 \\ \lambda_0 u_0 = \lambda_0 (r_0 - a) \\ \lambda_k u_k = \lambda_k (-a - \sum_{j=0}^{k-1} u_j + \sum_{j=0}^{k} r_j), k = 1, 2, ..., N-1 \end{cases} \tag{9}$$

Now we have $2N$ equations and $2N$ unknown variables. Therefore, all we need is to solve that system.

If there will be several solutions u^{opt*}, we need to select from them the one u^{opt} which minimizes the objective functional.

Numerical Part: numerical analysis of the formal problem using the facts obtained in the previous section

Let us try to solve this problem with substitution of the parameters with concrete values.

For simplicity, let us consider only two time intervals. Suppose then the parameters are

$$a = z_0 = 2, r = \begin{bmatrix} 6 \\ 7 \end{bmatrix}, \alpha = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \beta = \begin{bmatrix} 4 \\ 5 \end{bmatrix}, \gamma = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

So our goal is to find the optimal solution $u^{opt} = \begin{bmatrix} u_0 \\ u_1 \end{bmatrix}$. Also we need $\lambda = \begin{bmatrix} \lambda_0 \\ \lambda_1 \end{bmatrix}$. Let us substitute all the data to the system (9):

$$\begin{cases} \lambda_1 = 2 \cdot 5 \cdot u_1 \\ \lambda_0 = 2 \cdot 4 \cdot u_0 + 3 - \lambda_1 \\ \lambda_0 u_0 = \lambda_0 (6 - 2) \\ \lambda_1 u_1 = \lambda_1 (-2 + 6 + 7 - u_0) \end{cases} \Leftrightarrow \begin{cases} \lambda_1 = 10 u_1 \\ \lambda_0 = 8 u_0 + 3 - \lambda_1 \\ \lambda_0 u_0 = 4 \lambda_0 \\ \lambda_1 u_1 = \lambda_1 (11 - u_0) \end{cases}$$

This system has 4 solutions:

$$\left[\begin{cases} u_0 = -\frac{3}{8} \\ u_1 = 0 \\ \lambda_0 = 0 \\ \lambda_1 = 0 \end{cases} \begin{cases} u_0 = 4 \\ u_1 = 0 \\ \lambda_0 = 35 \\ \lambda_1 = 0 \end{cases} \begin{cases} u_0 = 4 \\ u_1 = 7 \\ \lambda_0 = -35 \\ \lambda_1 = 70 \end{cases} \begin{cases} u_0 = \frac{107}{18} \\ u_1 = \frac{91}{18} \\ \lambda_0 = 0 \\ \lambda_1 = \frac{455}{9} \end{cases}\right.$$

However, we also need to check:

- that $u_0, u_1 \in \mathscr{A} = \{u \in \mathbb{R}^N_+ \mid g(u) \leq 0\}$, where $g(u) = (-a - \sum_{i=0}^{k-1} u_i + \sum_{i=0}^{k-1} r_i), k = 1, \ldots, N$ from problem formulation
- that $\lambda_0, \lambda_1 \geq 0$ from KTT theorem formulation

Only answer $\begin{cases} u_0 = \frac{107}{18} \\ u_1 = \frac{91}{18} \\ \lambda_0 = 0 \\ \lambda_1 = \frac{455}{9} \end{cases}$ satisfies this conditions.

Therefore, $u^{opt} = \begin{bmatrix} \frac{107}{18} \\ \\ \frac{91}{18} \end{bmatrix}$ is global minimum and the solution for that particular problem.

Computational Studies

We decided to use python as programming language. We also used sympy library, which has functionaliti for solving systems of equations. This setup perfectly matches our requirements.

We use results from the previous sections as base for implementaion.

```
def J(U):
    global N, A, ALPHA, BETA, GAMMA, R

    total_cost = A*sum(ALPHA) + sum(GAMMA) + sum([b*u*u for b,u in zip(BETA, U)])
    u_r_differences = [u-r for u, r in zip(U, R)]
    for i in range(N):
        total_cost += ALPHA[i] * sum(u_r_differences[:i])
    return total_cost
```

```
def check_constraints(U):
    global N, A, R

    r_u_differences = [r-u for u, r in zip(U, R)]

    constraints_list = [(-A + sum(r_u_differences[:i+1])) for i in range(N)]

    return all([c <= 0 for c in constraints_list])
```

```
def check_positive(l):
    return all([x >=0 for x in l])
```

```
def check_conditions(U, L):
    return check_positive(U) and check_positive(L)  and check_constraints(U)
```

```
from sympy import symbols, Eq, solve

def construct_equations():
    global N, A, ALPHA, BETA, GAMMA, R

    us = symbols(", ".join([f"u{i}" for i in range(N)]))
    ls = symbols(", ".join([f"l{i}" for i in range(N)]))

    equations = []
    equations.append(Eq(ls[-1] , 2 * BETA[-1] * us[-1]))
    equations.append(Eq(ls[0]*us[0] , ls[0]*(R[0]-A)))

    for k in range(N-1):
        equations.append(Eq(ls[k] , 2 * BETA[k] * us[k] + sum(ALPHA[k+1:]) - sum(ls[k+1:])))
    for k in range(1, N):
        equations.append(Eq(ls[k]*us[k] , ls[k]*(-A - sum(us[:k]) + sum(R[:k+1]) )))

    return equations, us + ls
```

```
equations, variables =  construct_equations()
solutions = solve(equations, variables)
```

```
N = 2
A = 2
ALPHA = [2, 3]
BETA = [4, 5]
GAMMA = [1, 2]
R = [6, 7]

solutions = solve(equations, variables)
admirable_solutions = list(filter( lambda x: check_conditions(*x)  ,solutions))
optimal_solution, total_cost = min(map(lambda x: (x[0], J(x[0])), admirable_solutions), key = lambda x: x[1])

print(f"Optimal solution U: = {optimal_solution}\nwith total cost = {total_cost}\n")

# Optimal solution U: = (107/18, 91/18)
# with total cost = 10151/36
```

As you can see, the optimal solution is the same as we get in the previous section.

For more details check python notebook: https://github.com/Kadaverciant/EOAL

Conclusion

During this assignment we were able to solve given non-linear optimization problem using the technics and methods learned from Engineering Optimization and the Optimization-Based Approach for Machine Learning (EOAL) Spring 2023 course.

Our task happened to be convex optimization problem (we decided), so we were able to apply respective technics, such as Karush–Kuhn–Tucker (KKT) theorem.

As a result, we managed to provide an algorithm to solve proposed problem. Then we tried to check correctness by substituting parameters with custom values.

After all, we successfully implemented the derived algorithm using such instruments as python programming language and sympy library.

The program is able to solve this kind of task which we have demonstrated in this jupyter notebook. Despite our implementation is far from ideal, since it uses mostly naive methods while operating with mathematical structures, it still able to solve given problems and good for demonstration purposes.

References

- Slides provided by Prof. Dr. habil. Vadim Azhmyakov during the EOAL course.
- A. Beck ”Introduction to Nonlinear Optimization”, SIAM, Philadelphia, 2014

- https://docs.sympy.org/latest/index.html
- https://www.sciencedirect.com/science/article/pii/S2352146523000194?ref=pdf_download&fr=RR-2&rr=7c2c53446b26c429
- https://en.wikipedia.org/wiki/Karush–Kuhn–Tucker_conditions