Constrained Problem and Algorithm I

Jong-June Jeon

October 10, 2023

Department of Statistics, University of Seoul

Optimality

Proposition 1 (Global minimum of a convex function)

A local minimum of a convex function is the global minimum.

(proof) Let f be the objective function and x be the local minimum of f, and C be the feasible set. Since x is a local minimum of f, there is a ball B containing x with radius $R > 0$ such that

$$f(x) \leq f(z)$$

for all $z \in B$.

Let y be the global minimum then

$$f(x) > f(y).$$

Since $y \notin B$, $\|x - y\| > R$. We set $z = (1-\theta)x + \theta y$ with $\theta = R/(2\|y - x\|)$ then $z \in B$ (because $\|z - x\| = R/2$).

$$f(z) \underset{\text{convexity}}{\leq} (1-\theta)f(x) + \theta f(y) \underset{f(y)<f(x)}{<} (1-\theta)f(x) + \theta f(x) = f(x),$$

which is the contradiction of the definition of x, the local minima in B.

Proposition 2 (Characterization of local minimum)

Let f be a differentiable convex function on C and x the local minima of f, then

$$\nabla f(x)^\top (y - x) \geq 0$$

for all $y \in B$, where B is a ball in C.

(proof) Since f is convex on C,

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x)$$

for all $x, y \in C$.

Suppose that

$$\nabla f(x)^\top (y - x) < 0$$

for some $y \in B$. Let $z(t) = ty + (1-t)x$ where $t \in [0,1]$. Then $f(z(t))$ is a convex function on $[0,1]$. Then, the optimality condition in the 1-dimensional case,

$$\frac{\partial f(z(t))}{\partial t} = \nabla f(z(t))(y - x) = 0,$$

which is a contradiction.

(See figure 4.2 on p.139)

Consider the unconstrained convex problem:

Let $y(t) = x - t\nabla f(x)$, then it also holds that

$$\nabla f(x)^\top (y - x) \geq 0$$

by the convexity of f. That is,

$$\nabla f(x)^\top (y - x) = \nabla f(x)^\top (-t\nabla f(x)) = -t\|\nabla f(x)\|^2 \geq 0,$$

which implies that $\|\nabla f(x)\| = 0$.

Proposition 3 (optimality condition with equality conditions)

$$\begin{aligned} \min \quad & f(x) \\ \textit{subject to} \quad & Ax = b \end{aligned}$$

Let x *be a solution of the problem, then*

$$\nabla f(x)^\top (y - x) \geq 0$$

for all $y \in \{y : Ay = b\}$.

(proof) Since x is a solution in a feasible set ($Ax = b$), we can write $y = x + \nu$ where $\nu \in \mathcal{N}(A)$

So,

$$\nabla f(x)^\top (y - x) = \nabla f(x)^\top \nu$$

for all $\nu \in \mathcal{N}(A)$. If In other words,

Equivalent problem

Step 2

A constrained optimization problem may have different formulations that yield the same solution. In essence, these are referred to as equivalent problems, and a well-formulated equivalent problem can offer computational advantages to solve the problem. In some instances, it is possible to transform a constrained problem into an unconstrained one, while in other cases, even when dealing with a non-convex problem, an equivalent problem can exist that becomes a convex optimization problem. Therefore, in order to proficiently solve various forms of optimization problems, the knowledge to formulate equivalent problems is essential. This chapter focuses on studying general techniques for constructing equivalent problems in the context of optimization.

Standard form

$$
\begin{aligned}
\text{minimize} \quad & f_0(z) \\
\text{subject to} \quad & f_i(z) \leq 0 \text{ for } i = 1, \cdots, m \\
& h_j(z) = 0 \text{ for } j = 1, \cdots, p
\end{aligned}
$$

Box constraints

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & l_i \leq x_i \leq u_i \text{ for } j = 1, \cdots, p \end{aligned}$$

Then,

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & l_i - x_i \leq 0 \text{ for } j = 1, \cdots, m \\ & x_i - u_i \leq 0 \text{ for } j = 1, \cdots, m \end{aligned}$$

Maximization

$$\begin{aligned} \text{maximize} \quad & f_0(z) \\ \text{subject to} \quad & f_i(z) \leq 0 \text{ for } i = 1, \cdots, m \\ & h_j(z) = 0 \text{ for } j = 1, \cdots, p \end{aligned}$$

Then,

$$\begin{aligned} \text{minimize} \quad & -f_0(z) \\ \text{subject to} \quad & f_i(z) \leq 0 \text{ for } i = 1, \cdots, m \\ & h_j(z) = 0 \text{ for } j = 1, \cdots, p \end{aligned}$$

Change of variable

If $\phi : \mathbb{R}^n \mapsto \mathbb{R}^n$ is one-to-one and $D \subset dom(\phi)$ then

$$\begin{aligned} \text{minimize} \quad & f_0(\phi(z)) \\ \text{subject to} \quad & f_i(\phi(z)) \le 0 \text{ for } i = 1, \cdots, m \\ & h_j(\phi(z)) = 0 \text{ for } j = 1, \cdots, p \end{aligned}$$

Let $\tilde{f}_i(z) = f_i(\phi(z))$ and $\tilde{h}_i(z) = h_i(\phi(z))$ then the equivalent problem is given by

$$\begin{aligned} \text{minimize} \quad & \tilde{f}_0(z) \\ \text{subject to} \quad & \tilde{f}_i(z) \le 0 \text{ for } i = 1, \cdots, m \\ & \tilde{h}_j(z) = 0 \text{ for } j = 1, \cdots, p \end{aligned}$$

Slack variable $f_i(x) \leq 0$ if and only if there exists an $s_i \geq 0$ such that $f_i(x) + s_i = 0$. Thus, by introducing slack variables, we can rewrite the optimization problem with inequality constraints $f_i(x) \leq 0$ for $i = 1, \cdots, m$ as follows.

$$
\begin{aligned}
\text{minimize} \quad & f_0(z) \\
\text{subject to} \quad & s_i \geq 0 \text{ for } i = 1, \cdots, m \\
& f_i(z) + s_i = 0 \text{ for } i = 1, \cdots, m \\
& h_j(z) = 0 \text{ for } j = 1, \cdots, p
\end{aligned}
$$

Optimizing over some variables

$$
\begin{aligned}
\text{minimize} \quad & f_0(x_1, x_2) \\
\text{subject to} \quad & f_i(x_1) \le 0 \text{ for } i = 1, \cdots, m_1 \\
& \tilde{f}_i(x_2) \le 0 \text{ for } i = 1, \cdots, m_2
\end{aligned}
$$

Let $\tilde{f}_0(x_1) = \inf\{f_0(x_1, z) : \tilde{f}_i(z) \le 0, i = 1, \cdots, m\}$, then the equivalent problem is written by

$$
\begin{aligned}
\text{minimize} \quad & \tilde{f}_0(x_1) \\
\text{subject to} \quad & f_i(x_1) \le 0 \text{ for } i = 1, \cdots, m_1
\end{aligned}
$$

Examples 1

$$
\begin{aligned}
\text{minimize} \quad & x_1^\top P_{11} x_1 + 2x_1^\top P_{12} x_2 + x_2^\top P_{22} x_2 \\
\text{subject to} \quad & f_i(x_1) \le 0 \text{ for } i = 1, \cdots, m_1,
\end{aligned}
$$

where P_{11} and P_{12} are symmetric and positive definite. (See p 134.)

Let $g(x_2) = 2x_1^\top P_{12}x_2 + x_2^\top P_{22}x_2$ for a fixed x_1, then f is convex function. Since $\nabla g(x_2) = 2P_{12}^\top x_1 + 2P_{22}x_2$, the solution of $\nabla g(x_2) = 0$ gives

$$\inf_{x_2} x_1^\top P_{11}x_1 + 2x_1^\top P_{12}x_2 + x_2^\top P_{22}x_2 = x_1^\top (P_{11} - P_{12}P_{22}^{-1}P_{12}^\top)x_1.$$

Thus, the problem is reduced by

$$\begin{aligned} \text{minimize} \quad & x_1^\top (P_{11} - P_{12}P_{22}^{-1}P_{12}^\top)x_1 \\ \text{subject to} \quad & f_i(x_1) \le 0 \text{ for } i = 1, \cdots, m_1, \end{aligned}$$

Constrained Convex problem

Definition 2 (Linear Optimization problem)

$$\begin{aligned} \text{minimize} \quad & c^\top x + d \\ \text{subject to} \quad & Gx \leq h \\ & Ax = b, \end{aligned}$$

where $G \in \mathbb{R}^{m \times n}$ and $A \in \mathbb{R}^{p \times n}$

$$
\begin{aligned}
\min \quad & 12x_1 + 16x_2 \\
\text{subject to} \quad & -x_1 - 2x_2 \leq -40 \\
& -x_1 - x_2 \leq 30 \\
& -x_1 \leq 0,\ -x_2 \leq 0
\end{aligned}
$$

figure

Standard form of Linear programming

$$\begin{aligned} \text{minimize} \quad & c^\top x \\ \text{subject to} \quad & Ax = b \\ & x \geq 0. \end{aligned}$$

By simplex algorithm [Dantzig, 1947] the problem can be solved. But more efficient algorithms have been developed.

Equivalent problem 1 of the linear optimization in the definition

First,

$$\begin{aligned} \text{minimize} \quad & c^\top x + d \\ \text{subject to} \quad & Gx + s = h \\ & Ax = b, \\ & s \geq 0 \end{aligned}$$

s is called the slack variable.

Equivalent problem 2 of the linear optimization problem

Next let $x^+ = \max(x, 0)$ and $x^- = \max(-x, 0)$ then $x^+, x^- \geq 0$ and $x = x^+ - x^-$. Then,

$$
\begin{aligned}
\text{minimize} \quad & c^\top x^+ - c^\top x^- + d \\
\text{subject to} \quad & Gx^+ - Gx^- + s = h \\
& Ax^+ - Ax^- = b, \\
& s \geq 0, x^+ \geq 0, x^- \geq 0.
\end{aligned}
$$

Let $\tilde{c} = [c^\top, -c^\top]^\top$, $\tilde{b} = (h^\top, b^\top)^\top$, $\tilde{x} = (x^{+\top}, x^{-\top}, s)$ and

$$\tilde{A} = \begin{pmatrix} G & -G & I \\ A & -A & 0 \end{pmatrix}.$$

Then, the equivalent problem is written by the standard form,

$$\begin{aligned} \min \quad & \tilde{c}^\top \tilde{x} \\ \text{subject to} \quad & \tilde{A}\tilde{x} = \tilde{b}, \ \tilde{x} \geq 0 \end{aligned}$$

Implementation with python: cvxopt

Problem:

$$
\begin{aligned}
\text{minimize} \quad & 2x_1 + x_2 \\
\text{subject to} \quad & -3x_1 + x_2 \leq 1 \\
& x_1 + x_2 \geq 2 \\
& x_2 \geq 0 \\
& x_1 - 2x_2 \leq 4
\end{aligned}
$$

Implementation with python: cvxopt

Problem:

$$
\begin{aligned}
\text{minimize} \quad & 2x_1 + x_2 \\
\text{subject to} \quad & -3x_1 + x_2 && \leq 1 \\
& -x_1 - x_2 && \leq -2 \\
& -x_2 && \leq 0 \\
& x_1 - 2x_2 && \leq 4
\end{aligned}
$$

http://cvxopt.org/examples/tutorial/lp.html

```
from cvxopt import matrix, solvers
import numpy as np
c = matrix([2.0, 1.0])
G = matrix([[-3.0, -1.0, 0.0, 1.0], [1.0, -1.0, -1.0, -2.0]])
h = matrix([1.0, -2.0, 0.0, 4.0])
sol=solvers.lp(c,G,h)
type(sol)
print(sol["x"])
print(sol["s"])
print(sol["primal objective"])
```

Note that the function argument is solvers.lp(c,G,h,A,b) (see the equivalent problem 1).

Example 3 (Optimal transportation problem)

- The total supply of the product from warehouse i is a_i, where $i = 1, 2, \cdots, m$.
- The total demand for the product at outlet j is b_j , where $j = 1, 2, \cdots, n$.
- The cost of sending one unit of the product from warehouse i to outlet j is equal to c_{ij} , where $i = 1, 2, \cdots, m$ and $j = 1, 2, \cdots, n$.
- x_{ij} is the size of the shipment from warehouse i to outlet j and assume that the total cost of a shipment is linear in the size of the shipment.

The total outgoing shipment from the warehouse i is given by $\sum_{j=1}^{n} x_{ij}$, which should be equal to or less than b_i. The total supply to the outlet j is given by $\sum_{i=1} x_{ij}$, which is equal or greater than a_j

The optimal transportation problem is formulated as the following:

$$\begin{aligned} \min_{i,j} \quad & c_{ij}x_{ij} \\ \text{subject to} \quad & \sum_{j=1}^{m} x_{ij} \geq a_i, \forall i \\ & \sum_{i=1}^{m} x_{ij} \leq b_j \\ & x_{ij} \geq 0 \text{ for all } \forall i, j \end{aligned}$$

Note that the feasibility condition is $\sum_{j=1}^{n} b_j \leq \sum_{i=1}^{m} a_i$.

Example 4 (L_p-Wasserstein distance between discrete distribution)

Let $X \in \{x_i : 1 \leq i \leq k_1\}$ and $Y \in \{y_j : 1 \leq j \leq k_2\}$ be discrete random variables. Denote the distribution of X and Y by $\mathbf{p}$ and $\mathbf{q}$ where $p_i = \Pr(X = x_i)$ and $q_j = \Pr(Y = y_j)$. L_p-Wasserstein distance between p and q is defined by the following problem.

$$
\begin{aligned}
W_p(\mathbf{p}, \mathbf{q}) = \quad & \min_{\pi \in \mathbb{R}^{k_1 \times k_2}} \sum_{i,j} \pi_{ij} |x_i - y_j|^p \\
& \text{subject to} \sum_{j=1}^{k_2} \pi_{ij} = p_i, \ \forall i \\
& \sum_{i=1}^{k_1} \pi_{ij} = q_j, \ \forall j \\
& \pi_{ij} \geq 0, \ \forall i, j
\end{aligned}
$$

Definition 5 (Quadratic optimization with linear constraints)

$$\begin{aligned} \text{minimize} \quad & (1/2)x^\top P x + q^\top x + r \\ \text{subject to} \quad & Gx \leq h \\ & Ax = b \end{aligned}$$

Implementation in python

Problem:

$$
\begin{aligned}
\text{minimize} \quad & 2x_1^2 + x_2^2 + \frac{1}{2}x_1x_2 + x_1 + x_2 \\
\text{subject to} \quad & x_1 \geq 0 \\
& x_2 \geq 0 \\
& x_1 + x_2 = 1
\end{aligned}
$$

$$
\begin{aligned}
\text{minimize} \quad & \frac{1}{2}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}^\top \begin{pmatrix} 4 & 1/2 \\ 1/2 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix}^\top \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\
\text{subject to} \quad & \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \le 0 \\
& \begin{pmatrix} 1 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 1
\end{aligned}
$$

Implementation in python

http://cvxopt.org/examples/tutorial/qp.html

```
from cvxopt import matrix, solvers
Q = matrix([ [4.0, 0.5], [0.5, 1.0] ])
p = matrix([1.0, 1.0])
G = matrix([[-1.0,0.0],[0.0,-1.0]])
h = matrix([0.0,0.0])
A = matrix([1.0, 1.0], shape = (1,2))
b = matrix(1.0)
sol=solvers.qp(Q, p, G, h, A, b)
print(sol["x"])
```

Example 6 (Markowitz model)

Markowitz's portfolio model suggests optimality based on the variance of the return. The optimal portfolio is defined by the collection of assets with the minimum variance.

- $X = (X_1, \cdots, X_n)^\top$ is the return rate vector of n assets
- $\beta \in \mathbb{R}^n$: weight vector
- $\mu = (\mu_1, \cdots, \mu_n)^\top$ is the expected return rate vector of n assets.
- μ_0: target return rate.
- $X^\top \beta$ with $1^\top \beta = 1$: portfolio.

(Continue of the example)

Ideal objective function

$$
\begin{aligned}
\text{min} \quad & \mathsf{Var}(X^\top \beta) \\
\text{subject to} \quad & \mu^\top \beta = \mu_0 \\
& 1^\top \beta = 1 \\
& \beta \geq 0
\end{aligned}
$$

Note that $\mathrm{Var}(X^\top \beta) = \mathrm{E}((X^\top \beta)^2) - \mathrm{E}(X^\top \beta)^2 = \beta^\top \mathrm{E}(XX^\top)\beta - \mu_0^2$.

(Continue of the example)

Equivalent problem is

$$\begin{aligned} \text{min} \quad & \beta^\top \mathrm{E}(XX^\top)\beta \\ \text{subject to} \quad & \mu^\top \beta = \mu_0 \\ & 1^\top \beta = 1 \\ & \beta \geq 0 \end{aligned}$$

(Continue of the example)

Let $x_i = (x_{i1}, \cdots, x_{in})$ for $i = 1, \cdots, m$ and $\mathrm{E}(XX^\top)$ and μ are estimated based on the observations.

$$\begin{aligned} \text{min} \quad & \beta^\top \frac{1}{m} \sum_{i=1}^{m} x_i x_i^\top \beta \\ \text{subject to} \quad & \hat{\mu}^\top \beta = \mu_0 \\ & 1^\top \beta = 1 \\ & \beta \geq 0 \end{aligned}$$

Note that the problem can be solved by quadratic programming.

Definition 7 (Quadratic optimization with quadratic constraints)

$$\begin{aligned} \text{minimize} \quad & (1/2)x^\top P_0 x + q_0^\top x + r_0 \\ \text{subject to} \quad & (1/2)x^\top P_i x + q_i^\top x + r_i \leq 0 \text{ for } i = 1, \cdots, m \\ & Ax = b, \end{aligned}$$

where $P_i \in S_+^n$.

This problem is called QCQP (See p 153). This problem can be solved by Second-order cone programming (SOCP), which will be shown.

Example 8 (Optimal allocation 1 [Phan-huy Hao, 1982])

Let a city be divided into m districts, let its summarized location vector $a_j \in \mathbb{R}^2$, and let w_j be the frequency by which the fire department will be called upon in this district. Find the location $x \in \mathbb{R}^2$ for a fire station. In case of fire breaking out in any district j, the frequency-weighted maximal distance to be covered from the fire station is minimal. So, in case of fire breaking out in any district j, the frequency-weighted maximal distance to be covered from the fire station is minimal.

The objective function is define by

$$\min_{x \in \mathbb{R}^2} \max_{j=1,\cdots,m} w_j^2 \|x - a_j\|^2.$$

This optimization problem is written by QCQP:

$$\begin{array}{ll} \min\limits_{d \in \mathbb{R}, x \in \mathbb{R}^2} & d \\ \text{subject to} & w_j^2 \|x - a_j\|^2 \leq d, \text{ for } j = 1, \cdots, m \end{array}$$

Example 9 (Max-cut problem)

Let $G = (V, E)$ be an undirected graph. For $U \subset V$, let $\delta(U) = \{uv \in E : u \in U, v \in U\}$. Here, $\delta(U)$ is called the cut determined by vertex set U. The Max-cut problem is to solve $\max\{|\delta(U)| : U \subset V\}$. This is NP-hard. Instead, we will be content to find a cut that is sufficiently large.

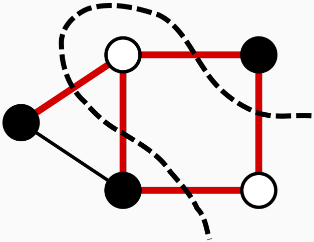

Figure 1: an example of Max-cut [1]

[1] https://en.wikipedia.org/wiki/Maximum_cut

(Continue of the example)

Let $x_i = 1$ if $i \in U$ and $x_i = -1$ otherwise. Then the Max-cut problem is solved by

$$\begin{aligned} \max \quad & \frac{1}{2}\sum_{i<j} w_{ij}(1 - x_i x_j) \\ & |x_i| = 1 \ \forall i, \end{aligned}$$

where w_{ij} is a weight of the edge (i, j). Note that $w_{ij} = w_{ji}$ if $(i, j) \in E$. Assume that W is symmetric and positive definite where $(W)_{ij} = w_{ij}$. This problem is relaxed by

$$\begin{aligned} \max \quad & \frac{1}{2}\sum_{i<j} w_{ij}(1 - x_i x_j) \\ & x_i^2 = 1 \ \forall i, \end{aligned}$$

(Continue of the example)

Since w_{ij} is symmetric and $x_i^2 = 1$ for all i, the equivalent problem is written by

$$\begin{aligned} \min \quad & \frac{1}{2}\sum_{i,j} x_i w_{ij} x_j \\ & x_i^2 = 1 \ \forall i, \end{aligned}$$

That is,

$$\begin{aligned} \min \quad & \frac{1}{2} x^\top W x \\ & \|x\|^2 = 1. \end{aligned}$$

Definition 10 (Second-order cone problem)

$$
\begin{aligned}
\text{minimize} \quad & f^\top x \\
\text{subject to} \quad & \|A_i x + b_i\|_2 \leq c_i^\top x + d_i \text{ for } i = 1, \cdots, m \\
& Fx = g,
\end{aligned}
$$

where $x \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n_i \times n}$, and $F \in \mathbb{R}^{p \times n}$

Second-order cone: $\{(x, t) \in \mathbb{R}^k \times \mathbb{R} : \|x\| \leq t\}$

QCQP: formulation by SOCP

$$
\begin{aligned}
\text{minimize} \quad & (1/2)x^\top P_0 x + q_0^\top x + r_0 \\
\text{subject to} \quad & (1/2)x^\top P_i x + q_i^\top x + r_i \le 0 \text{ for } i = 1, \cdots, m \\
& Fx = g,
\end{aligned}
$$

Since $P_i \in S_+^n$, we can write $P_i = E_i D_i E_i^\top = (D_i^{1/2} E_i^\top)^\top (D_i^{1/2} E_i^\top)$ by eigen-decomposition. Let $P_i^{1/2} = D_i^{1/2} E_i^\top$ then the QCQP has equivalent to the following problem:

$$
\begin{aligned}
\text{minimize} \quad & \frac{1}{2}\|P_0^{1/2}x + P_0^{-1/2}q_0\|^2 + r_0 - \frac{1}{2}q_0^\top P_0^{-1}q_0 \\
\text{subject to} \quad & \frac{1}{2}\|P_i^{1/2}x + P_i^{-1/2}q_i\|^2 + r_i - \frac{1}{2}q_i^\top P_i^{-1}q_i \le 0 \text{ for } i = 1, \cdots, m \\
& Fx = g,
\end{aligned}
$$

Then,

$$
\begin{aligned}
\text{minimize} \quad & t \\
\text{subject to} \quad & \|P_0^{1/2}x + P_0^{-1/2}q_0\| \le t \\
& \|P_i^{1/2}x + P_i^{-1/2}q_i\| \le \sqrt{q_i^\top P_i^{-1}q_i - r_i} \text{ for } i = 1, \cdots, m \\
& Fx = g.
\end{aligned}
$$

Note that P_i and $P_i^{1/2}$ are known.

Implementation in python

$$
\begin{array}{ll}
\text{minimize} & -2x_1 + x_2 + 5x_3 \\
\text{subject to} & \left\| \begin{bmatrix} -13x_1 + 3x_2 + 5x_3 - 3 \\ -12x_1 + 12x_2 - 6x_3 - 2 \end{bmatrix} \right\| \leq -12x_1 - 6x_2 + 5x_3 - 12 \\
& \left\| \begin{bmatrix} -3x_1 + 6x_2 + 2x_3 \\ x_1 + 9x_2 + 2x_3 + 3 \\ -x_1 - 19x_2 + 3x_3 - 42 \end{bmatrix} \right\| \leq -3x_1 + 6x_2 - 10x_3 + 27
\end{array}
$$

Implementation in python

https://www.cvxpy.org/examples/basic/socp.html

```
from cvxopt import matrix, solvers
c = matrix([-2., 1., 5.])
G = [ matrix( [[12., 13., 12.], [6., -3., -12.], [-5., -5., 6.]] ) ]
G += [ matrix( [[3., 3., -1., 1.], [-6., -6., -9., 19.], [10., -2., -2., -3.]] ) ]
h = [ matrix( [-12., -3., -2.] ),  matrix( [27., 0., 3., -42.] ) ]
sol = solvers.socp(c, Gq = G, hq = h)
sol["status"]
print(sol["x"])
print(sol["zq"][0])
print(sol["zq"][1])
```

Definition 11 (Semidefinite program)

$$\begin{aligned} \min \quad & c^\top x \\ \text{subject to} \quad & x_1 F_1 + \cdots + x_n F_n + G \preceq 0 \\ & Ax = b, \end{aligned}$$

where $G, F_1, \cdots, F_n \in \mathcal{S}^k$ and $A \in \mathbb{R}^{m \times n}$.

Here, $x_1 F_1 + \cdots + x_n F_n + G \preceq 0$ means that $-(x_1 F_1 + \cdots + x_n F_n + G)$ is nonnegative definite.

Definition 12 (Standard form of Semidefinite programs)

$$\begin{aligned} \min_X \quad & \text{tr}(CX) \\ \text{subject to} \quad & \text{tr}(A_i X) = b_i, \ \text{for} i = 1, \cdots, p \\ & X \succeq 0, \end{aligned}$$

where $X, C, A_1, \cdots, A_p, \in \mathcal{S}^n$,

HW

- KOSPI 200 주가데이터를 모으고 마코위츠 모형을 적합하여라. 평균 수익률은 실행가능영역에서 각자 정하여라.
- 세로축을 평균로그 수익률, 가로축을 포트폴리오의 표준편차로 하는 효율적 경계를 도출하여라.

Phan-huy Hao, E. (1982).
Quadratically constrained quadratic programming: Some applications and a method for solution.
Zeitschrift für Operations Research, 26:105–119.