Convex Optimization
Homework 3

Spring 1400
Due date: 13th of Farvardin

---

1. Consider the optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x_1, x_2) \\ \text{subject to} & 2x_1 + x_2 \geq 1 \\ & x_1 + 3x_2 \geq 1 \\ & x_1 \geq 0, \quad x_2 \geq 0 \end{array}$$

Make a sketch of the feasible set. For each of the following objective functions, give the optimal set and the optimal value.

(a) $f_0(x_1, x_2) = x_1 + x_2$.
(b) $f_0(x_1, x_2) = -x_1 - x_2$.
(c) $f_0(x_1, x_2) = x_1$.
(d) $f_0(x_1, x_2) = \max\{x_1, x_2\}$.
(e) $f_0(x_1, x_2) = x_1^2 + 9x_2^2$.

2. Consider the optimization problem

$$\text{minimize} \quad f_0(x) = -\sum_{i=1}^{m} \log(b_i - a_i^T x)$$

with domain $\mathbf{dom} f_0 = \{x \mid Ax \prec b\}$, where $A \in \mathbb{R}^{m \times n}$ (with rows a_i^T). We assume that $\mathbf{dom} f_0$ is nonempty. Prove the following facts (which include the results quoted without proof on page 141 from the book).

(a) $\mathbf{dom} f_0$ is unbounded if and only if there exists a $v \neq 0$ with $Av \preceq 0$.
(b) f_0 is unbounded below if and only if there exists a v with $Av \preceq 0$, $Av \neq 0$. Hint. There exists a v such that $Av \preceq 0$, $Av \neq 0$ if and only if there exists no $z \succ 0$ such that $A^T z = 0$. This follows from the theorem of alternatives in example 2.21, page 50 of the book.
(c) If f_0 is bounded below then its minimum is attained, i.e., there exists an x that satisfies the optimality condition (equation 4.23 from the book).
(d) The optimal set is affine: $X_{opt} = \{x^* + v \mid Av = 0\}$, where x^* is any optimal point.

3. We consider the selection of n nonnegative activity levels, denoted $x_1, \ldots, x_n$. These activities consume m resources, which are limited. Activity j consumes $A_{ij}x_j$ of resource i, where A_{ij} are given. The total resource consumption is additive, so the total of resource i consumed is $c_i = \sum_{j=1}^{n} A_{ij}x_j$. (Ordinarily we have $A_{ij} \geq 0$, i.e., activity j consumes resource i. But we allow the possibility that $A_{ij} < 0$, which means that activity j actually generates resource i as a by-product.) Each resource consumption is limited: we must have $c_i \leq c_{max}$, where c_{max} are given. Each activity generates revenue, which is a piecewise-linear concave function of the activity level:

$$r_j(x_j) = \begin{cases} p_j x_j & 0 \leq x_j \leq q_j \\ p_j q_j + p_j^{disc}(x_j - q_j) & x_j \geq q_j \end{cases}$$

Here $p_j > 0$ is the basic price, $q_j > 0$ is the quantity discount level, and p_j^{disc} is the j disc quantity discount price, for (the product of) activity j. (We have $0 < p_j^{disc} < p_j$.) The total revenue is the sum of the revenues associated with each activity, i.e., $\sum_{j=1}^{n} r_j(x_j)$. The goal is to choose activity levels that maximize the total revenue while respecting the resource limits. Show how to formulate this problem as an LP.

4. Use CVX, CVXPY, or Convex.jl to verify the optimal values you obtained (analytically) for problem #1.

5. Each of the following CVX code fragments describes a convex constraint on the scalar variables `x`, `y`, and `z`, but violates the CVX rule set, and so is invalid. Briefly explain why each fragment is invalid. Then, rewrite each one in an equivalent form that conforms to the CVX rule set. In your reformulations, you can use linear equality and inequality constraints, and inequalities constructed using CVX functions. You can also introduce additional variables, or use LMIs. Be sure to explain (briefly) why your reformulation is equivalent to the original constraint, if it is not obvious.

   Check your reformulations by creating a small problem that includes these constraints, and solving it using CVX. Your test problem doesn't have to be feasible; it's enough to verify that CVX processes your constraints without error.

   Remark. This looks like a problem about 'how to use CVX software', or 'tricks for using CVX'. But it really checks whether you understand the various composition rules, convex analysis, and constraint reformulation rules.

   (a) `norm([x + 2*y, x - y]) == 0.`
   (b) `square(square(x + y)) <= x - y.`
   (c) `1/x + 1/y <= 1; x >= 0; y >= 0.`
   (d) `norm([max(x,1), max(y,2)]) <= 3*x + y.`
   (e) `x*y >= 1; x >= 0; y >= 0.`
   (f) `(x + y)^2/sqrt(y) <= x - y + 5.`
   (g) `x^3 + y^3 <= 1; x >= 0; y >= 0.`
   (h) `x + z <= 1 + sqrt(x*y - z^2); x >= 0; y >= 0.`

6. In lecture 1 we encountered the function

$$f(p) = \max_{i=1,\ldots,n} |\log a_i^T p - \log I_{des}|$$

   where $a_i \in \mathbb{R}^m$,and $I_{des} > 0$ are given,and $p \in \mathbb{R}_+^m$.

   (a) Show that $\exp f$ is convex on $\{p \mid a_i^T p > 0, i = 1, \ldots, n\}$.
   (b) Show that the constraint 'no more than half of the total power is in any 10 lamps' is convex (i.e., the set of vectors p that satisfy the constraint is convex).
   (c) Show that the constraint 'no more than half of the lamps are on' is (in general) not convex.

7. Formulate the following optimization problems as semidefinite programs. The variable is $x \in \mathbb{R}^n$; $F(x)$ is defined as

$$F(x) = F_0 + x_1 F_1 + x_2 F_2 + \cdots + x_n F_n$$

   with $F_i \in \mathbb{S}^m$ .The domain of f in each subproblem is $\mathbf{dom} f = \{x \in \mathbb{R}^n \mid F(x) \succ 0\}$.

   (a) Minimize $f(x) = c^T F(x)^{-1} c$ where $c \in \mathbb{R}^m$.
   (b) Minimize $f(x) = \max_{i=1,\ldots,K} c_i^T F(x)^{-1} c_i$ where $c_i \in \mathbb{R}^m$, $i = 1, \ldots, K$.
   (c) Minimize $f(x) = \sup_{\|c\|_2 \leq 1} c^T F(x)^{-1} c$.
   (d) Minimize $f(x) = \mathbb{E}[c^T F(x)^{-1} c]$ where c is a random vector with mean $\mathbb{E}[c] = \bar{c}$ and covariance $\mathbb{E}[(c - \bar{c})(c - \bar{c})^T] = S$.

8. We consider a linear dynamical system with state $x(t) \in \mathbb{R}^n$, $t = 0, \dots, N$, and actuator or input signal $u(t) \in \mathbb{R}$, for $t = 0, \dots, N-1$. The dynamics of the system is given by the linear recurrence

$$x(t+1) = Ax(t) + bu(t), \qquad t = 0, \dots, N-1$$

where $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$ are given. We assume that the initial state is zero, i.e., $x(0) = 0$. The *minimum fuel optimal control problem* is to choose the inputs $u(0), \dots, u(N-1)$ so as to minimize the total fuel consumed, which is given by

$$F = \sum_{t=0}^{N-1} f(u(t))$$

subject to the constraint that $x(N) = x_{des}$, where N is the (given) time horizon, and $x_{des} \in \mathbb{R}^n$ is the (given) desired final or target state. The function $f : \mathbb{R} \to \mathbb{R}$ is the fuel use map for the actuator, and gives the amount of fuel used as a function of the actuator signal amplitude. In this problem we use

$$f(a) = \begin{cases} |a| & |a| \leq 1 \\ 2|a| - 1 & |a| > 1 \end{cases}$$

This means that fuel use is proportional to the absolute value of the actuator signal, for actuator signals between -1 and 1; for larger actuator signals the marginal fuel efficiency is half.

(a) Formulate the minimum fuel optimal control problem as an LP.

(b) Solve the minimum fuel optimal control problem described in exercise 4.16 of Convex Optimization, for the instance with problem data

$$A = \begin{bmatrix} -1 & 0.4 & 0.8 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \qquad b = \begin{bmatrix} 1 \\ 0 \\ 0.3 \end{bmatrix}, \qquad x_{des} = \begin{bmatrix} 7 \\ 2 \\ -6 \end{bmatrix}, \qquad N = 30.$$

You can do this by forming the LP you found in **(a)**, or more directly using CVX. Plot the actuator signal $u(t)$ as a function of time t.

9. In a *Boolean linear program*, the variable x is constrained to have components equal to zero or one:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \preceq b \\ & x_i \in \{0, 1\}, \quad i = 1, \dots, n \end{array}$$

In general, such problems are very difficult to solve, even though the feasible set is finite (containing at most 2^n points). In a general method called relaxation, the constraint that x_i be zero or one is replaced with the linear inequalities $0 \leq x_i \leq 1$:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \preceq b \\ & 0 \leq x_i \leq 1, \quad i = 1, \dots, n \end{array}$$

We refer to this problem as the LP relaxation of the Boolean LP. The LP relaxation is far easier to solve than the original Boolean LP.

(a) Show that the optimal value of the LP relaxation is a lower bound on the optimal value of the Boolean LP. What can you say about the Boolean LP if the LP relaxation is infeasible?

(b) It sometimes happens that the LP relaxation has a solution with $x_i \in \{0, 1\}$. What can you say in this case?

(c) Let p^* be the optimal value of the Boolean LP problem and x^{rlx} be a solution to the LP relaxation problem. So $L = c^T x^{rlx}$ is a lower bound on p^*. The relaxed solution x^{rlx} can also be used to guess a Boolean point $\hat{x}$, by rounding its entries, based on a threshold $t \in [0, 1]$:

$$\hat{x}_i = \begin{cases} 1 & x_i^{rlx} \geq t \\ 0 & o.w., \end{cases}$$

for $i = 1, \ldots, n$. Evidently $\hat{x}$ is Boolean (i.e., has entries in $\{0, 1\}$). If it is feasible for the Boolean LP, i.e., if $A\hat{x} \preceq b$, then it can be considered a guess at a good, if not optimal, point for the Boolean LP. Its objective value, $U = c^T\hat{x}$, is an upper bound on p^*. If U and L are close, then $\hat{x}$ is nearly optimal; specifically, $\hat{x}$ cannot be more than $(U - L)$-suboptimal for the Boolean LP.

This rounding need not work; indeed, it can happen that for all threshold values, $\hat{x}$ is infeasible. But for some problem instances, it can work well.

Of course, there are many variations on this simple scheme for (possibly) constructing a feasible, good point from x^{rlx}.

Finally, we get to the problem. Generate problem data using one of the following.

Matlab code:

```
rand('state;,0);
n=100;
m=300;
A=rand(m,n);
b=A*ones(n,1)/2;
c=-rand(n,1);
```

Python code:

```
import numpy as np
np.random.seed(0)
(m, n) = (300, 100)
A = np.random.rand(m, n); A = np.asmatrix(A)
b = A.dot(np.ones((n, 1)))/2; b = np.asmatrix(b)
c = -np.random.rand(n, 1); c = np.asmatrix(c)
```

Julia code:

```
srand(0);
n=100;
m=300;
A=rand(m,n);
b=A*ones(n,1)/2;
c=-rand(n,1);
```

You can think of x_i as a job we either accept or decline, and $-c_i$ as the (positive) revenue we generate if we accept job i. We can think of $Ax \preceq b$ as a set of limits on m resources. A_{ij}, which is positive, is the amount of resource i consumed if we accept job j; b_i, which is positive, is the amount of resource i available.

Find a solution of the relaxed LP and examine its entries. Note the associated lower bound L. Carry out threshold rounding for (say) 100 values of t, uniformly spaced over $[0, 1]$. For each value of t, note the objective value $c^T\hat{x}$ and the maximum constraint violation $max_i(A\hat{x} - b)_i$. Plot the objective value and the maximum violation versus t. Be sure to indicate on the plot the values of t for which $\hat{x}$ is feasible, and those for which it is not.

Find a value of t for which $\hat{x}$ is feasible, and gives minimum objective value, and note the associated upper bound U. Give the gap $U - L$ between the upper bound on p^* and the lower bound on p^*.

In Matlab, if you define vectors `obj` and `maxviol`, you can find the upper bound as `U=min(obj(find(maxviol<=0)))`.

Good luck, and happy Nowruz!