EE364a Homework 4 solutions

4.1 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) $f_0(x_1, x_2) = x_1 + x_2$.

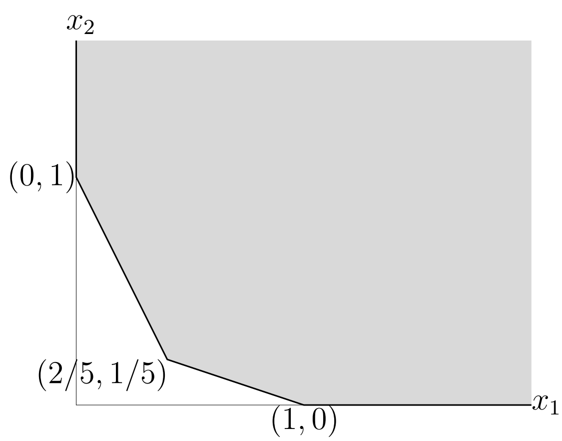
(b) $f_0(x_1, x_2) = -x_1 - x_2$.

(c) $f_0(x_1, x_2) = x_1$.

(d) $f_0(x_1, x_2) = \max\{x_1, x_2\}$.

(e) $f_0(x_1, x_2) = x_1^2 + 9x_2^2$.

Solution. The feasible set is shown in the figure.



(a) $x^\star = (2/5, 1/5)$.

(b) Unbounded below.

(c) $X_{\text{opt}} = \{(0, x_2) \mid x_2 \geq 1\}$.

(d) $x^\star = (1/3, 1/3)$.

(e) $x^\star = (1/2, 1/6)$. This is optimal because it satisfies $2x_1 + x_2 = 7/6 > 1$, $x_1 + 3x_2 = 1$, and

$$\nabla f_0(x^\star) = (1, 3)$$

is perpendicular to the line $x_1 + 3x_2 = 1$.

4.8 *Some simple LPs.* Give an explicit solution of each of the following LPs.

(a) *Minimizing a linear function over an affine set.*

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b. \end{array}$$

Solution. We distinguish three possibilities.

- The problem is infeasible ($b \notin \mathcal{R}(A)$). The optimal value is ∞.
- The problem is feasible, and c is orthogonal to the nullspace of A. We can decompose c as

$$c = A^T \lambda + \hat{c}, \qquad A\hat{c} = 0.$$

  ($\hat{c}$ is the component in the nullspace of A; $A^T\lambda$ is orthogonal to the nullspace.) If $\hat{c} = 0$, then on the feasible set the objective function reduces to a constant:

$$c^T x = \lambda^T A x + \hat{c}^T x = \lambda^T b.$$

  The optimal value is $\lambda^T b$. All feasible solutions are optimal.
- The problem is feasible, and c is not in the range of A^T ($\hat{c} \neq 0$). The problem is unbounded ($p^\star = -\infty$). To verify this, note that $x = x_0 - t\hat{c}$ is feasible for all t; as t goes to infinity, the objective value decreases unboundedly.

In summary,

$$p^\star = \begin{cases} +\infty & b \notin \mathcal{R}(A) \\ \lambda^T b & c = A^T\lambda \text{ for some } \lambda \\ -\infty & \text{otherwise.} \end{cases}$$

(b) *Minimizing a linear function over a halfspace.*

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a^T x \leq b, \end{array}$$

where $a \neq 0$.

Solution. This problem is always feasible. The vector c can be decomposed into a component parallel to a and a component orthogonal to a:

$$c = a\lambda + \hat{c},$$

with $a^T\hat{c} = 0$.

- If $\lambda > 0$, the problem is unbounded below. Choose $x = -ta$, and let t go to infinity:

$$c^T x = -tc^T a = -t\lambda a^T a \rightarrow -\infty$$

  and

$$a^T x - b = -ta^T a - b \leq 0$$

  for large t, so x is feasible for large t. Intuitively, by going very far in the direction $-a$, we find feasible points with arbitrarily negative objective values.

- If $\hat{c} \neq 0$, the problem is unbounded below. Choose $x = \frac{b}{a^T a} a - t\hat{c}$ and let t go to infinity.
- If $c = a\lambda$ for some $\lambda \leq 0$, the optimal value is λb; any point x with $a^T x = b$ is optimal.

In summary, the optimal value is

$$p^\star = \begin{cases} \lambda b & c = a\lambda \text{ for some } \lambda \leq 0 \\ -\infty & \text{otherwise.} \end{cases}$$

(c) *Minimizing a linear function over a rectangle.*

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & l \preceq x \preceq u, \end{array}$$

where l and u satisfy $l \preceq u$.

Solution. The objective and the constraints are separable: The objective is a sum of terms $c_i x_i$, each dependent on one variable only; each constraint depends on only one variable. We can therefore solve the problem by minimizing over each component of x independently. The optimal $x_i^\star$ minimizes $c_i x_i$ subject to the constraint $l_i \leq x_i \leq u_i$. If $c_i > 0$, then $x_i^\star = l_i$; if $c_i < 0$, then $x_i^\star = u_i$; if $c_i = 0$, then any x_i in the interval $[l_i, u_i]$ is optimal. Therefore, the optimal value of the problem is

$$p^\star = l^T c^+ - u^T c^-,$$

where $c_i^+ = \max\{c_i, 0\}$ and $c_i^- = \max\{-c_i, 0\}$.

(d) *Minimizing a linear function over the probability simplex.*

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \mathbf{1}^T x = 1, \quad x \succeq 0. \end{array}$$

What happens if the equality constraint is replaced by an inequality $\mathbf{1}^T x \leq 1$?

We can interpret this LP as a simple portfolio optimization problem. The vector x represents the allocation of our total budget over different assets, with x_i the fraction invested in asset i. The return of each investment is fixed and given by $-c_i$, so our total return (which we want to maximize) is $-c^T x$. If we replace the budget constraint $\mathbf{1}^T x = 1$ with an inequality $\mathbf{1}^T x \leq 1$, we have the option of not investing a portion of the total budget.

Solution. Suppose the components of c are sorted in increasing order with

$$c_1 = c_2 = \cdots = c_k < c_{k+1} \leq \cdots \leq c_n.$$

We have

$$c^T x \geq c_1(\mathbf{1}^T x) = c_{\min}$$

for all feasible x, with equality if and only if

$$x_1 + \cdots + x_k = 1, \qquad x_1 \geq 0, \ldots, x_k \geq 0, \qquad x_{k+1} = \cdots = x_n = 0.$$

We conclude that the optimal value is $p^\star = c_1 = c_{\min}$. In the investment interpretation this choice is quite obvious. If the returns are fixed and known, we invest our total budget in the investment with the highest return.

If we replace the equality with an inequality, the optimal value is equal to

$$p^\star = \min\{0, c_{\min}\}.$$

(If $c_{\min} \leq 0$, we make the same choice for x as above. Otherwise, we choose $x = 0$.)

4.11 *Problems involving ℓ_1- and ℓ_∞-norms.* Formulate the following problems as LPs. Explain in detail the relation between the optimal solution of each problem and the solution of its equivalent LP.

(a) Minimize $\|Ax - b\|_\infty$ (ℓ_∞-norm approximation).

(b) Minimize $\|Ax - b\|_1$ (ℓ_1-norm approximation). In each problem, $A \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$ are given.

Solution.

(a) Equivalent to the LP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & Ax - b \preceq t\mathbf{1} \\ & Ax - b \succeq -t\mathbf{1}. \end{array}$$

in the variables x, t. To see the equivalence, assume x is fixed in this problem, and we optimize only over t. The constraints say that

$$-t \leq a_k^T x - b_k \leq t$$

for each k, *i.e.*, $t \geq |a_k^T x - b_k|$, *i.e.*,

$$t \geq \max_k |a_k^T x - b_k| = \|Ax - b\|_\infty.$$

Clearly, if x is fixed, the optimal value of the LP is $p^\star(x) = \|Ax - b\|_\infty$. Therefore optimizing over t and x simultaneously is equivalent to the original problem.

(b) Equivalent to the LP

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T s \\ \text{subject to} & Ax - b \preceq s \\ & Ax - b \succeq -s. \end{array}$$

Assume x is fixed in this problem, and we optimize only over s. The constraints say that

$$-s_k \leq a_k^T x - b_k \leq s_k$$

for each k, *i.e.*, $s_k \geq |a_k^T x - b_k|$. The objective function of the LP is separable, so we achieve the optimum over s by choosing

$$s_k = |a_k^T x - b_k|,$$

and obtain the optimal value $p^\star(x) = \|Ax - b\|_1$. Therefore optimizing over t and s simultaneously is equivalent to the original problem.

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

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

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 $0 \leq x_i \leq 1$:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \preceq b \\ & 0 \leq x_i \leq 1, \quad i = 1, \ldots, n. \end{array} \tag{2}$$

We refer to this problem as the *LP relaxation* of the Boolean LP (1). The LP relaxation is far easier to solve than the original Boolean LP.

(a) Show that the optimal value of the LP relaxation (2) is a lower bound on the optimal value of the Boolean LP (1). 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?

Solution.

(a) The feasible set of the relaxation includes the feasible set of the Boolean LP. It follows that the Boolean LP is infeasible if the relaxation is infeasible, and that the optimal value of the relaxation is less than or equal to the optimal value of the Boolean LP.

(b) The optimal solution of the relaxation is also optimal for the Boolean LP.

4.43 *Eigenvalue optimization via SDP.* Suppose $A : \mathbf{R}^n \to \mathbf{S}^m$ is affine, *i.e.*,

$$A(x) = A_0 + x_1 A_1 + \cdots + x_n A_n$$

where $A_i \in \mathbf{S}^m$. Let $\lambda_1(x) \geq \lambda_2(x) \geq \cdots \geq \lambda_m(x)$ denote the eigenvalues of $A(x)$. Show how to pose the following problems as SDPs.

(a) Minimize the maximum eigenvalue $\lambda_1(x)$.

(b) Minimize the spread of the eigenvalues, $\lambda_1(x) - \lambda_m(x)$.

Solution.

(a) We use the property that $\lambda_1(x) \leq t$ if and only if $A(x) \preceq tI$. We minimize the maximum eigenvalue by solving the SDP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & A(x) \preceq tI. \end{array}$$

The variables are $x \in \mathbf{R}^n$ and $t \in \mathbf{R}$.

(b) $\lambda_1(x) \leq t_1$ if and only if $A(x) \preceq t_1 I$ and $\lambda_m(A(x)) \geq t_2$ if and only if $A(x) \succeq t_2 I$, so we can minimize $\lambda_1 - \lambda_m$ by solving

$$\begin{array}{ll} \text{minimize} & t_1 - t_2 \\ \text{subject to} & t_2 I \preceq A(x) \preceq t_1 I. \end{array}$$

This is an SDP with variables $t_1 \in \mathbf{R}$, $t_2 \in \mathbf{R}$, and $x \in \mathbf{R}^n$.

4.50 *Bi-criterion optimization.* Figure 4.11 shows the optimal trade-off curve and the set of achievable values for the bi-criterion optimization problem

$$\text{minimize (w.r.t. } \mathbf{R}^2_+) \quad (\|Ax - b\|^2, \|x\|_2^2),$$

for some $A \in \mathbf{R}^{100 \times 10}$, $b \in \mathbf{R}^{100}$. Answer the following questions using information from the plot. We denote by x_{ls} the solution of the least-squares problem

$$\text{minimize} \quad \|Ax - b\|_2^2.$$

(a) What is $\|x_{\text{ls}}\|_2$?

(b) What is $\|Ax_{\text{ls}} - b\|_2$?

(c) What is $\|b\|_2$?

(d) Give the optimal value of the problem

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|_2^2 \\ \text{subject to} & \|x\|_2^2 = 1. \end{array}$$

(e) Give the optimal value of the problem

$$\begin{array}{ll} \text{minimize} & \|Ax-b\|_2^2 \\ \text{subject to} & \|x\|_2^2 \le 1. \end{array}$$

(f) Give the optimal value of the problem

$$\text{minimize } \|Ax-b\|_2^2 + \|x\|_2^2.$$

(g) What is the rank of A?

Solution.

(a) $\|x_{\rm ls}\|_2 = 3$.

(b) $\|Ax_{\rm ls} - b\|_2^2 = 2$.

(c) $\|b\|_2 = \sqrt{10}$.

(d) About 5.

(e) About 5.

(f) About $3+4$.

(g) $\textbf{rank}\, A = 10$, since the LS solution is unique.

4.59 *Robust optimization.* In some optimization problems there is uncertainty or variation in the objective and constraint functions, due to parameters or factors that are either beyond our control or unknown. We can model this situation by making the objective and constraint functions $f_0, \ldots, f_m$ functions of the optimization variable $x \in \mathbf{R}^n$ *and* a parameter vector $u \in \mathbf{R}^k$ that is unknown, or varies. In the *stochastic optimization* approach, the parameter vector u is modeled as a random variable with a known distribution, and we work with the expected values $\mathbf{E}_u\, f_i(x,u)$. In the *worst-case analysis* approach, we are given a set U that u is known to lie in, and we work with the maximum or worst-case values $\sup_{u \in U} f_i(x,u)$. To simplify the discussion, we assume there are no equality constraints.

(a) *Stochastic optimization.* We consider the problem

$$\begin{array}{ll} \text{minimize} & \mathbf{E}\, f_0(x,u) \\ \text{subject to} & \mathbf{E}\, f_i(x,u) \le 0, \quad i = 1, \ldots, m, \end{array}$$

where the expectation is with respect to u. Show that if f_i are convex in x for each u, then this stochastic optimization problem is convex.

(b) *Worst-case optimization.* We consider the problem

$$\begin{array}{ll} \text{minimize} & \sup_{u \in U} f_0(x,u) \\ \text{subject to} & \sup_{u \in U} f_i(x,u) \le 0, \quad i = 1, \ldots, m. \end{array}$$

Show that if f_i are convex in x for each u, then this worst-case optimization problem is convex.

(c) *Finite set of possible parameter values.* The observations made in parts (a) and (b) are most useful when we have analytical or easily evaluated expressions for the expected values $\mathbf{E}\, f_i(x,u)$ or the worst-case values $\sup_{u \in U} f_i(x,u)$.

Suppose we are given the set of possible values of the parameter is finite, *i.e.*, we have $u \in \{u_1, \ldots, u_N\}$. For the stochastic case, we are also given the probabilities of each value: $\mathbf{prob}(u = u_i) = p_i$, where $p \in \mathbf{R}^N$, $p \succeq 0$, $\mathbf{1}^T p = 1$. In the worst-case formulation, we simply take $U \in \{u_1, \ldots, u_N\}$.

Show how to set up the worst-case and stochastic optimization problems explicitly (*i.e.*, give explicit expressions for $\sup_{u \in U} f_i$ and $\mathbf{E}_u\, f_i$).

Solution.

(a) Follows from the fact that the inequality

$$f_i(\theta x + (1-\theta)y, u) \leq \theta f(x,u) + (1-\theta) f(y,u)$$

is preserved when we take expectations on both sides.

(b) If $f_i(x,u)$ is convex in x for fixed u, then $\sup_u f_i(x,u)$ is convex in x.

(c) Stochastic formulation:

$$\begin{array}{ll} \text{minimize} & \sum_i p_k f_0(x, u_k) \\ \text{subject to} & \sum_k p_k f_i(x, u_k) \leq 0, \quad i = 1, \ldots, m. \end{array}$$

Worst-case formulation:

$$\begin{array}{ll} \text{minimize} & \max_k f_0(x, u_k) \\ \text{subject to} & \max_k f_i(x, u_k) \leq 0, \quad i = 1, \ldots, m. \end{array}$$

A3.2 *'Hello World' in CVX.* Use CVX to verify the optimal values you obtained (analytically) for exercise 4.1 in *Convex Optimization.*

Solution.

(a) $p^\star = 0.6$

(b) $p^\star = -\infty$

(c) $p^\star = 0$

(d) $p^\star = \frac{1}{3}$

(e) $p^\star = \frac{1}{2}$

```
%exercise 4.1 using CVX

%set up a vector to store optimal values of problems
optimal_values=zeros(5,1);
```

```
%part a
cvx_begin
variable x(2)
minimize(x(1)+x(2))
2*x(1)+x(2) >= 0;
x(1)+3*x(2) >= 1;
x >= 0;
cvx_end

optimal_values(1)=cvx_optval;

%part b
cvx_begin
variable x(2)
minimize(-sum(x))
2*x(1)+x(2) >= 0;
x(1)+3*x(2) >= 1;
x >= 0;
cvx_end
optimal_values(2)=cvx_optval;

%part c
cvx_begin
variable x(2)
minimize(x(1))
2*x(1)+x(2) >= 0;
x(1)+3*x(2) >= 1;
x >= 0;
cvx_end
optimal_values(3)=cvx_optval;

%part d
cvx_begin
variable x(2)
minimize(max(x))
2*x(1)+x(2) >= 0;
x(1)+3*x(2) >= 1;
x >= 0;
cvx_end
optimal_values(4)=cvx_optval;
```

```
%part e
cvx_begin
variable x(2,1)
minimize( square(x(1))+ 9*square(x(2)) )
2*x(1)+x(2) >= 0;
x(1)+3*x(2) >= 1;
x >= 0;
cvx_end
optimal_values(5)=cvx_optval;
```

A3.3 *Reformulating constraints in CVX.* 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`

Solution.

(a) The lefthand side is correctly identified as convex, but equality constraints are only valid with affine left and right hand sides. Since the norm of a vector is zero if and only if the vector is zero, we can express the constraint as `x + 2*y == 0; x - y == 0`, or simply `x == 0; y == 0`.

(b) The problem is that `square()` can only accept affine arguments, because it is convex, but not increasing. To correct this use `square_pos()` instead:

```
square_pos(square(x + y)) <= x - y
```

We can also reformulate this constraint by introducing an additional variable.

```
variable t
square(x + y) <= t
square(t) <= x - y
```

Note that, in general, decomposing the objective by introducing new variables doesn't need to work. It works in this case because the outer `square` function is convex and monotonic over $\mathbf{R}_+$.

Alternatively, we can rewrite the constraint as

```
(x + y)^4 <= x - y
```

(c) $1/x$ isn't convex, unless you restrict the domain to $\mathbf{R}_{++}$. We can write this one as `inv_pos(x) + inv_pos(y) <= 1`. The `inv_pos` function has domain $\mathbf{R}_{++}$ so the constraints $x > 0$, $y > 0$ are (implicitly) included.

(d) The problem is that `norm()` can only accept affine argument since it is convex but not increasing. One way to correct this is to introduce new variables `u` and `v`:

```
norm([u, v]) <= 3*x + y
max(x,1) <= u
max(y,2) <= v
```

Decomposing the objective by introducing new variables works here because `norm` is convex and monotonic over $\mathbf{R}_+^2$, and in particular over $[1, \infty) \times [2, \infty)$.

(e) xy isn't concave, so this isn't going to work as stated. But we can express the constraint as `x >= inv_pos(y)`. (You can switch around `x` and `y` here.) Another solution is to write the constraint as `geo_mean([x, y]) >= 1`. We can also give an LMI representation:

```
[x 1; 1 y] == semidefinite(2)
```

(f) This fails when we attempt to divide a convex function by a concave one. We can write this as

```
quad_over_lin(x + y, sqrt(y)) <= x - y + 5
```

This works because `quad_over_lin` is monotone decreasing in the second argument, so it can accept a concave function here, and `sqrt` is concave.

(g) The function $x^3 + y^3$ is convex for $x \geq 0$, $y \geq 0$. But x^3 isn't convex for $x < 0$, so CVX is going to reject this statement. One way to rewrite this constraint is

```
quad_pos_over_lin(square(x),x) + quad_pos_over_lin(square(y),y) <= 1
```

This works because `quad_pos_over_lin` is convex and increasing in its first argument, hence accepts a convex function in its first argument. (The function `quad_over_lin`, however, is not increasing in its first argument, and so won't work.)

Alternatively, and more simply, we can rewrite the constraint as

```
pow_pos(x,3) + pow_pos(y,3) <= 1
```

(h) The problem here is that xy isn't concave, which causes CVX to reject the statement. To correct this, notice that

$$\sqrt{xy - z^2} = \sqrt{y(x - z^2/y)},$$

so we can reformulate the constraint as

```
x + z <= 1 + geo_mean([x - quad_over_lin(z,y), y])
```

This works, since `geo_mean` is concave and nondecreasing in each argument. It therefore accepts a concave function in its first argument.

We can check our reformulations by writing the following feasibility problem in CVX (which is obviously infeasible)

```
cvx_begin
    variables x y u v z
    x == 0;
    y == 0;
    (x + y)^4 <= x - y;
```

```
    inv_pos(x) + inv_pos(y)  <= 1;
    norm([u; v]) <= 3*x + y;
    max(x,1) <= u;
    max(y,2) <= v;
    x >= inv_pos(y);
    x >= 0;
    y >= 0;
    quad_over_lin(x + y, sqrt(y)) <= x - y + 5;
    pow_pos(x,3) + pow_pos(y,3) <= 1;
    x+z <= 1+geo_mean([x-quad_over_lin(z,y), y])
cvx_end
```