EE364a Homework 3 solutions

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

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

  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 $x = \frac{b}{\|a\|_2^2} a - t\hat{c}$ and let t go to infinity.
- If $c = a\lambda$ for some $\lambda \leq 0$, the optimal value is $c^T \left(\frac{b}{\|a\|_2^2} a \right) = \lambda b$.

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 $x = 0$.)

(e) *Minimizing a linear function over a unit box with a total budget constraint.*

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

where α is an integer between 0 and n. What happens if α is not an integer (but satisfies $0 \leq \alpha \leq n$)? What if we change the equality to an inequality $\mathbf{1}^T x \leq \alpha$?

Solution. We first consider the case of integer α. Suppose

$$c_1 \leq \cdots \leq c_{i-1} < c_i = \cdots = c_\alpha = \cdots = c_k < c_{k+1} \leq \cdots \leq c_n.$$

The optimal value is

$$c_1 + c_2 + \cdots + c_\alpha$$

i.e., the sum of the smallest α elements of c. x is optimal if and only if

$$x_1 = \cdots = x_{i-1} = 1, \qquad x_i + \cdots + x_k = \alpha - i + 1, \qquad x_{k+1} = \cdots = x_n = 0.$$

If α is not an integer, the optimal value is

$$p^\star = c_1 + c_2 + \cdots + c_{\lfloor \alpha \rfloor} + c_{1+\lfloor \alpha \rfloor}(\alpha - \lfloor \alpha \rfloor).$$

In the case of an inequality constraint $\mathbf{1}^T x \leq \alpha$, with α an integer between 0 and n, the optimal value is the sum of the α smallest nonpositive coefficients of c.

(f) *Minimizing a linear function over a unit box with a weighted budget constraint.*

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

with $d \succ 0$, and $0 \le \alpha \le \mathbf{1}^T d$.

Solution. We make a change of variables $y_i = d_i x_i$, and consider the problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^n (c_i/d_i) y_i \\ \text{subject to} & \mathbf{1}^T y = \alpha, \quad 0 \preceq y \preceq d. \end{array}$$

Suppose the ratios c_i/d_i have been sorted in increasing order:

$$\frac{c_1}{d_1} \le \frac{c_2}{d_2} \le \cdots \le \frac{c_n}{d_n}.$$

To minimize the objective, we choose

$$y_1 = d_1, \quad y_2 = d_2, \quad \ldots, \quad y_k = d_k,$$

$$y_{k+1} = \alpha - (d_1 + \cdots + d_k), \qquad y_{k+2} = \cdots = y_n = 0,$$

where $k = \max\{i \in \{1, \ldots, n\} \mid d_1 + \cdots + d_i \le \alpha\}$ (and $k = 0$ if $d_1 > \alpha$). In terms of the original variables,

$$x_1 = \cdots = x_k = 1, \qquad x_{k+1} = (\alpha - (d_1 + \cdots + d_k))/d_{k+1}, \qquad x_{k+2} = \cdots = x_n = 0.$$

The optimal value is

$$p^\star = c_1 + c_2 + \cdots + c_k + (c_{k+1}/d_{k+1})(\alpha - (d_1 + \cdots + d_k)).$$

4.17 *Optimal activity levels.* 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 $A_{ij} \ge 0$, *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 $c_i \le c_i^{\max}$, where $c_i^{\max}$ are given. Each activity generates revenue, which is a piecewise-linear concave function of the activity level:

$$r_j(x_j) = \begin{cases} p_j x_j & 0 \le x_j \le q_j \\ p_j q_j + p_j^{\text{disc}}(x_j - q_j) & x_j \ge q_j. \end{cases}$$

Here $p_j > 0$ is the basic price, $q_j > 0$ is the quantity discount level, and p_j^{disc} is the quantity discount price, for (the product of) activity j. (We have $0 < p_j^{\text{disc}} < p_j$.) 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.

Solution. The basic problem can be expressed as

$$\begin{array}{ll} \text{maximize} & \sum_{j=1}^{n} r_j(x_j) \\ \text{subject to} & x \succeq 0 \\ & Ax \preceq c^{\max}. \end{array}$$

This is a convex optimization problem since the objective is concave and the constraints are a set of linear inequalities. To transform it to an equivalent LP, we first express the revenue functions as

$$r_j(x_j) = \min\{p_j x_j, p_j q_j + p_j^{\text{disc}}(x_j - q_j)\},$$

which holds since r_j is concave. It follows that $r_j(x_j) \geq u_j$ if and only if

$$p_j x_j \geq u_j, \qquad p_j q_j + p_j^{\text{disc}}(x_j - q_j) \geq u_j.$$

We can form an LP as

$$\begin{array}{ll} \text{maximize} & \mathbf{1}^T u \\ \text{subject to} & x \succeq 0 \\ & Ax \preceq c^{\max} \\ & p_j x_j \geq u_j, \quad p_j q_j + p_j^{\text{disc}}(x_j - q_j) \geq u_j, \quad j = 1, \ldots, n, \end{array}$$

with variables x and u.

To show that this LP is equivalent to the original problem, let us fix x. The last set of constraints in the LP ensure that $u_i \leq r_i(x)$, so we conclude that for every feasible x, u in the LP, the LP objective is less than or equal to the total revenue. On the other hand, we can always take $u_i = r_i(x)$, in which case the two objectives are equal.

4.58 *Optimal consumption.* In this problem we consider the optimal way to consume (or spend) an initial amount of money (or other asset) k_0 over time. The variables are $c_1, \ldots, c_T$, where $c_t \geq 0$ denotes the *consumption* in period t. The utility derived from a consumption level c is given by $u(c)$, where $u : \mathbf{R} \to \mathbf{R}$ is an increasing concave function. The present value of the utility derived from the consumption is given by

$$U = \sum_{t=1}^{T} \beta^t u(c_t),$$

where $0 < \beta < 1$ is a *discount factor.*

Let k_t denote the amount of money available for investment in period t. We assume that it earns an investment return given by $f(k_t)$, where $f : \mathbf{R} \to \mathbf{R}$ is an increasing, concave *investment return function*, which satisfies $f(0) = 0$. For example if the funds earn simple interest at rate R percent per period, we have $f(a) = (R/100)a$. The amount to be consumed, *i.e.*, c_t, is withdrawn at the end of the period, so we have the recursion

$$k_{t+1} = k_t + f(k_t) - c_t, \quad t = 0, \ldots, T.$$

The initial sum $k_0 > 0$ is given. We require $k_t \geq 0$, $t = 1, \ldots, T+1$ (but more sophisticated models, which allow $k_t < 0$, can be considered).

Show how to formulate the problem of maximizing U as a convex optimization problem. Explain how the problem you formulate is equivalent to this one, and exactly how the two are related.

Hint. Show that we can replace the recursion for k_t given above with the inequalities

$$k_{t+1} \leq k_t + f(k_t) - c_t, \quad t = 0, \ldots, T.$$

(Interpretation: the inequalities give you the option of throwing money away in each period.) For a more general version of this trick, see exercise 4.6.

Solution. We start with the problem

$$\begin{array}{ll} \text{maximize} & U = \sum_{t=1}^{T} \beta^t u(c_t) \\ \text{subject to} & k_{t+1} = k_t + f(k_t) - c_t, \ t = 0, \ldots, T \\ & k_t \geq 0, \ t = 1, \ldots, T+1, \end{array}$$

with variables $c_1, \ldots, c_T$ and $k_1, \ldots, k_{T+1}$. The objective is concave, since it is a positive weighted sum of concave functions. But the budget recursion constraints are *not* convex, since they are *equality* constraints involving the (possibly) nonlinear function f. The hint explains what to do: we look instead at the modified problem

$$\begin{array}{ll} \text{maximize} & U = \sum_{t=1}^{T} \beta^t u(c_t) \\ \text{subject to} & k_{t+1} \leq k_t + f(k_t) - c_t, \ t = 0, \ldots, T \\ & k_t \geq 0, \ t = 1, \ldots, T+1. \end{array}$$

This problem *is* convex, since the budget inequalities can be written as

$$k_{t+1} - k_t - f(k_t) + c_t \leq 0,$$

where the lefthand side is a convex function of the variables c and k.

We will now show that when we solve the modified problem with the inequality constraints, for any optimal solution we actually get *equality* for each of the budget constraints. This means that the solution of the modified problem is actually optimal for the original problem as well. To see this, we note that by changing the equality constraints into inequalities, we are *relaxing* the constraints (*i.e.*, making them looser), and therefore, if anything, we improve the objective compared to the original problem.

Let $c^\star$ and $k^\star$ be optimal for the modified problem. Suppose that at some period s, we have

$$k^\star_{s+1} < k^\star_s + f(k^\star_s) - c^\star_s.$$

This looks pretty suspicious, since it means that in period s, we are actually throwing away money (*i.e.*, we are not investing or consuming all of our available funds). Now consider a new consumption stream $\tilde{c}$ defined as

$$\tilde{c}_t = \begin{cases} c_t^\star & t \neq s \\ c_t^\star + \epsilon & t = s \end{cases}$$

where ϵ is a small positive number such that

$$k_{s+1}^\star \leq k_s^\star + f(k_s^\star) - c_s^\star$$

holds. In words, $\tilde{c}$ is the same consumption stream as $c^\star$, except in the period when we throw away some money (in $c^\star$) we just consume a little more. Clearly we have $U(\tilde{c}) \geq U(c^\star)$, since the two streams consume the same amount for every period except one, in which we consume more with $\tilde{c}$. (Here we use the fact that U is increasing.)

Let $\tilde{k}$ be the asset stream that results from the consumption stream $\tilde{c}$. Then all the constraints of the original problem are satisfied for $\tilde{c}$ and $\tilde{k}$, and yet $c^\star$ has a lower objective value than $\tilde{c}$. That contradicts optimality of $c^\star$. We conclude that for $c^\star$, we have

$$k_{t+1}^\star = k_t^\star + f(k_t^\star) - c_t^\star.$$

Solutions to additional exercises

1. *Circularly symmetric Huber function.* The scalar Huber function is defined as

$$f_{\rm hub}(x) = \begin{cases} (1/2)x^2 & |x| \leq 1 \\ |x| - 1/2 & |x| > 1. \end{cases}$$

This convex function comes up in several applications, including robust estimation. This problem concerns generalizations of the Huber function to $\mathbf{R}^n$. One generalization to $\mathbf{R}^n$ is given by $f_{\rm hub}(x_1) + \cdots + f_{\rm hub}(x_n)$, but this function is not circularly symmetric, *i.e.*, invariant under transformation of x by an orthogonal matrix. A generalization to $\mathbf{R}^n$ that *is* circularly symmetric is

$$f_{\rm cshub}(x) = f_{\rm hub}(\|x\|) = \begin{cases} (1/2)\|x\|_2^2 & \|x\|_2 \leq 1 \\ \|x\|_2 - 1/2 & \|x\|_2 > 1. \end{cases}$$

(The subscript stands for 'circularly symmetric Huber function'.) Show that $f_{\rm cshub}$ is convex. Find the conjugate function $f^*_{\rm cshub}$.

Solution. We can't directly use the composition form given above, since $f_{\rm hub}$ is *not* nondecreasing. But we can write $f_{\rm cshub} = h \circ g$, where $h : \mathbf{R} \to \mathbf{R}$ and $g : \mathbf{R}^n \to \mathbf{R}$ are defined as

$$h(x) = \begin{cases} 0 & x \leq 0 \\ x^2/2 & 0 \leq x \leq 1 \\ x - 1/2 & x > 1, \end{cases}$$

and $g(x) = \|x\|_2$. We can think of g as as a version of the scalar Huber function, modified to be zero when its argument is negative. Clearly, g is convex and $\tilde{h}$ is convex and increasing. Thus, from the composition rules we conclude that $f_{\rm cshub}$ is convex.

Now we will show that

$$f^*_{\rm cshub}(y) = \begin{cases} (1/2)\|y\|_2^2 & \|y\|_2 \leq 1 \\ \infty & \text{otherwise.} \end{cases}$$

Suppose $\|y\|_2 > 1$. Taking $x = ty/\|y\|_2$, we see that for $t \geq 1$ we have

$$y^T x - f(x) = t\|y\|_2 - t + 1/2 = t(\|y\|_2 - 1) + 1/2.$$

Letting $t \to \infty$, we see that for any y with $\|y\|_2 > 1$, $\sup_x (y^T x - f(x)) = \infty$. Thus, $f^*_{\rm cshub}(y) = \infty$ for $\|y\|_2 > 1$.

Now suppose $\|y\|_2 \leq 1$. We can write $f^*_{\rm cshub}(y)$ as

$$f^*_{\rm cshub}(y) = \max \left\{ \sup_{\|x\|_2 \leq 1} (y^T x - (1/2)\|x\|_2^2), \sup_{\|x\|_2 \geq 1} (y^T x - \|x\|_2 + 1/2) \right\}.$$

It is easy to show that $y^T x - (1/2)\|x\|_2^2$ is maximized over $\{x \mid \|x\|_2 \le 1\}$ when $x = y$ (set the gradient of $y^T x - (1/2)\|x\|_2^2$ equal to zero). This gives

$$\sup_{\|x\|_2 \le 1} (y^T x - (1/2)\|x\|_2^2) = (1/2)\|y\|_2^2.$$

To find $\sup_{\|x\|_2 \ge 1}(y^T x - \|x\|_2 + 1/2)$, notice that for $\|x\|_2 \ge 1$

$$y^T x - \|x\|_2 + 1/2 \le \|y\|_2\|x\|_2 - \|x\|_2 + 1/2 = \|x\|_2(\|y\|_2 - 1) + 1/2 \le \|y\|_2 - 1/2.$$

Here, the first inequality follows from Cauchy-Schwarz, and the second inequality follows from $\|y\|_2 \le 1$ and $\|x\|_2 \ge 1$. Furthermore, if we choose $x = y/\|y\|_2$, then

$$y^T x - \|x\|_2 + 1/2 = \|y\|_2 - 1/2,$$

thus,

$$\sup_{\|x\|_2 \ge 1} (y^T x - \|x\|_2 + 1/2) = \|y\|_2 - 1/2.$$

For $\|y\|_2 \le 1$

$$\sup_{\|x\|_2 \ge 1} (y^T x - \|x\|_2 + 1/2) = \|y\|_2 - 1/2 \le (1/2)\|y\|_2^2 = \sup_{\|x\|_2 \le 1} (y^T x - (1/2)\|x\|_2^2),$$

so we conclude that for $\|y\|_2 \le 1$, $f^*_{\text{cshub}}(y) = (1/2)\|y\|_2^2$.

2. *Minimizing a function over the probability simplex.* Find simple necessary and sufficient conditions for $x \in \mathbf{R}^n$ to minimize a differentiable convex function f over the probability simplex, $\{x \mid \mathbf{1}^T x = 1,\ x \succeq 0\}$.

Solution. The simple basic optimality condition is that x is feasible, *i.e.*, $x \succeq 0$, $\mathbf{1}^T x = 1$, and that $\nabla f(x)^T (y - x) \ge 0$ for all feasible y. We'll first show this is equivalent to

$$\min_{i=1,\ldots,n} \nabla f(x)_i \ge \nabla f(x)^T x.$$

To see this, suppose that $\nabla f(x)^T (y - x) \ge 0$ for all feasible y. Then in particular, for $y = e_i$, we have $\nabla f(x)_i \ge \nabla f(x)^T x$, which is what we have above. To show the other way, suppose that $\nabla f(x)_i \ge \nabla f(x)^T x$ holds, for $i = 1, \ldots, n$. Let y be feasible, *i.e.*, $y \succeq 0$, $\mathbf{1}^T y = 1$. Then multiplying $\nabla f(x)_i \ge \nabla f(x)^T x$ by y_i and summing, we get

$$\sum_{i=1}^n y_i \nabla f(x)_i \ge \left(\sum_{i=1}^n y_i\right) \nabla f(x)^T x = \nabla f(x)^T x.$$

The lefthand side is $y^T \nabla f(x)$, so we have $\nabla f(x)^T (y - x) \ge 0$.

Now we can simplify even further. The condition above can be written as

$$\min_{i=1,\ldots,n} \frac{\partial f}{\partial x_i} \ge \sum_{i=1}^n x_i \frac{\partial f}{\partial x_i}.$$

But since $\mathbf{1}^T x = 1$, $x \succeq 0$, we have

$$\min_{i=1,\ldots,n} \frac{\partial f}{\partial x_i} \leq \sum_{i=1}^{n} x_i \frac{\partial f}{\partial x_i},$$

and it follows that

$$\min_{i=1,\ldots,n} \frac{\partial f}{\partial x_i} = \sum_{i=1}^{n} x_i \frac{\partial f}{\partial x_i}.$$

The right hand side is a mixture of $\partial f/\partial x_i$ terms and equals the minimum of all of the terms. This is possible only if $x_k = 0$ whenever $\partial f/\partial x_k > \min_i \partial f/\partial x_i$.

Thus we can write the (necessary and sufficient) optimality condition as $\mathbf{1}^T x = 1$, $x \succeq 0$, and, for each k,

$$x_k > 0 \;\Rightarrow\; \frac{\partial f}{\partial x_k} = \min_{i=1,\ldots,n} \frac{\partial f}{\partial x_i}.$$

In particular, for k's with $x_k > 0$, $\partial f/\partial x_k$ are all equal.

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 descreasing 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
```

4. *Optimal activity levels.* Solve the optimal activity level problem described in exercise 4.17 in *Convex Optimization*, for the instance with problem data

$$A = \begin{bmatrix} 1 & 2 & 0 & 1 \\ 0 & 0 & 3 & 1 \\ 0 & 3 & 1 & 1 \\ 2 & 1 & 2 & 5 \\ 1 & 0 & 3 & 2 \end{bmatrix}, \quad c^{\max} = \begin{bmatrix} 100 \\ 100 \\ 100 \\ 100 \\ 100 \end{bmatrix}, \quad p = \begin{bmatrix} 3 \\ 2 \\ 7 \\ 6 \end{bmatrix}, \quad p^{\text{disc}} = \begin{bmatrix} 2 \\ 1 \\ 4 \\ 2 \end{bmatrix}, \quad q = \begin{bmatrix} 4 \\ 10 \\ 5 \\ 10 \end{bmatrix}.$$

You can do this by forming the LP you found in your solution of exercise 4.17, or more directly, using `cvx`. Give the optimal activity levels, the revenue generated by each one, and the total revenue generated by the optimal solution. Also, give the average price per unit for each activity level, *i.e.*, the ratio of the revenue associated with an activity, to the activity level. (These numbers should be between the basic and discounted prices for each activity.) Give a *very brief* story explaining, or at least commenting on, the solution you find.

Solution: The following Matlab/CVX code solves the problem. (Here we write the problem in a form close to its original statement, and let CVX do the work of reformulating it as an LP!)

```
A = [1 2 0 1;
     0 0 3 1;
     0 3 1 1;
     2 1 2 5;
     1 0 3 2];

cmax = [100;100;100;100;100];
p = [3;2;7;6];
pdisc = [2;1;4;2];
q = [4;10;5;10];

cvx_begin
  variable x(4)
  maximize(sum(min(p.*x,p.*q+pdisc.*(x-q))))
  subject to
    x >= 0;
    A*x <= cmax
cvx_end

x
r = min(p.*x,p.*q+pdisc.*(x-q))
totr = sum(r)
avgPrice = r./x
```

The result of the code is

```
x =

    4.0000
   22.5000
   31.0000
    1.5000


r =

   12.0000
   32.5000
  139.0000
    9.0000


totr =

  192.5000


avgPrice =

    3.0000
    1.4444
    4.4839
    6.0000
```

We notice that the 3rd activity level is the highest and is also the one with the highest basic price. Since it also has a high discounted price its activity level is higher than the discount quantity level and it produces the highest contribution to the total revenue. The 4th activity has a discounted price which is substantially lower then the basic price and its activity is therefore lower than the discount quantity level. Moreover it requires the use of a lot of resources and therefore its activity level is low.

5. *The illumination problem.* This exercise concerns the illumination problem described in lecture 1 (pages 9–11). We'll take $I_{\mathrm{des}} = 1$ and $p_{\max} = 1$, so the problem is

$$\begin{array}{ll} \text{minimize} & f_0(p) = \max_{k=1,\ldots,n} |\log(a_k^T p)| \\ \text{subject to} & 0 \leq p_j \leq 1, \quad j = 1, \ldots, m, \end{array} \tag{1}$$

with variable $p \in \mathbf{R}^n$. You will compute several approximate solutions, and compare the results to the exact solution, for a specific problem instance.

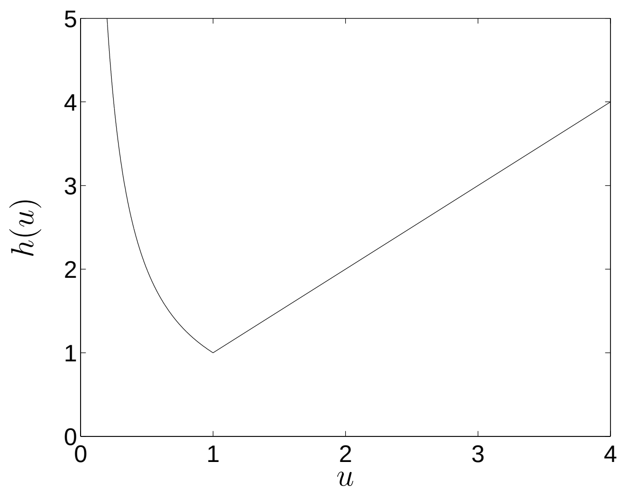
As mentioned in the lecture, the problem is equivalent to

$$\begin{array}{ll} \text{minimize} & \max_{k=1,\ldots,n} h(a_k^T p) \\ \text{subject to} & 0 \leq p_j \leq 1, \quad j = 1, \ldots, m, \end{array} \tag{2}$$

where $h(u) = \max\{u, 1/u\}$ for $u > 0$. The function h, shown in the figure below, is nonlinear, nondifferentiable, and convex. To see the equivalence between (1) and (2), we note that

$$\begin{aligned} f_0(p) &= \max_{k=1,\ldots,n} |\log(a_k^T p)| \\ &= \max_{k=1,\ldots,n} \max\{\log(a_k^T p), \log(1/a_k^T p)\} \\ &= \log \max_{k=1,\ldots,n} \max\{a_k^T p, 1/a_k^T p\} \\ &= \log \max_{k=1,\ldots,n} h(a_k^T p), \end{aligned}$$

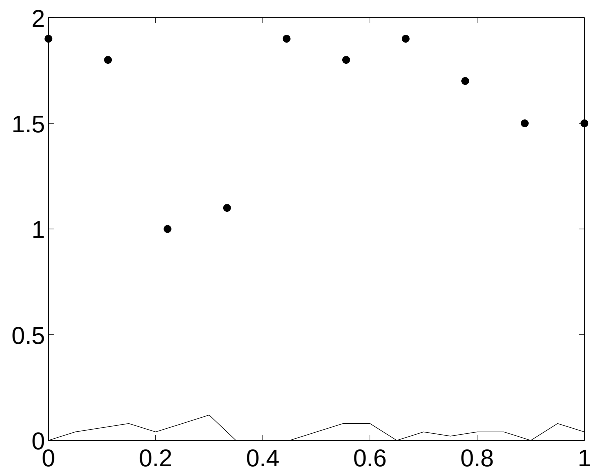
and since the logarithm is a monotonically increasing function, minimizing f_0 is equivalent to minimizing $\max_{k=1,\ldots,n} h(a_k^T p)$.



The problem instance. The specific problem data are for the geometry shown below, using the formula

$$a_{kj} = r_{kj}^{-2} \max\{\cos\theta_{kj}, 0\}$$

from the lecture. There are 10 lamps ($m = 10$) and 20 patches ($n = 20$). We take $I_{\text{des}} = 1$ and $p_{\max} = 1$. The problem data are given in the file `illum_data.m` on the course website. Running this script will construct the matrix A (which has rows a_k^T), and plot the lamp/patch geometry as shown below.



Equal lamp powers. Take $p_j = \gamma$ for $j = 1, \ldots, m$. Plot $f_0(p)$ versus γ over the interval $[0, 1]$. Graphically determine the optimal value of γ, and the associated objective value.

You can evaluate the objective function $f_0(p)$ in Matlab as `max(abs(log(A*p)))`.

Least-squares with saturation. Solve the least-squares problem

$$\text{minimize} \quad \sum_{k=1}^{n} (a_k^T p - 1)^2 = \|Ap - \mathbf{1}\|_2^2.$$

If the solution has negative values for some p_i, set them to zero; if some values are greater than 1, set them to 1. Give the resulting value of $f_0(p)$.

Least-squares solutions can be computed using the Matlab backslash operator: `A\b` returns the solution of the least-squares problem

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

Regularized least-squares. Solve the regularized least-squares problem

$$\text{minimize} \quad \sum_{k=1}^{n} (a_k^T p - 1)^2 + \rho \sum_{j=1}^{m} (p_j - 0.5)^2 = \|Ap - \mathbf{1}\|_2^2 + \rho \|p - (1/2)\mathbf{1}\|_2^2,$$

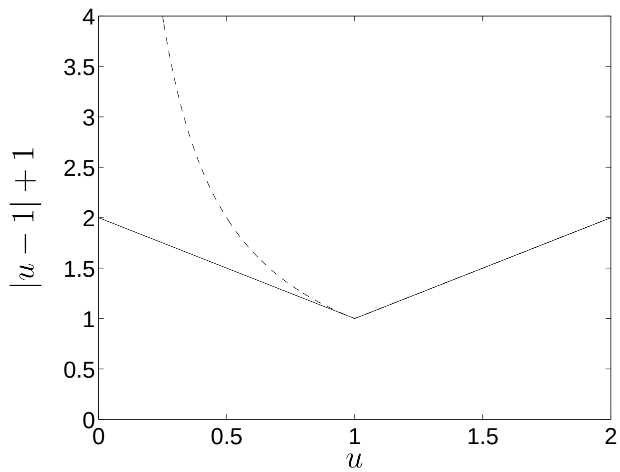
where $\rho > 0$ is a parameter. Increase ρ until all coefficients of p are in the interval $[0, 1]$. Give the resulting value of $f_0(p)$.

You can use the backslash operator in Matlab to solve the regularized least-squares problem.

Chebyshev approximation. Solve the problem

$$\begin{array}{ll} \text{minimize} & \max_{k=1,\ldots,n} |a_k^T p - 1| = \|Ap - \mathbf{1}\|_\infty \\ \text{subject to} & 0 \leq p_j \leq 1, \quad j = 1, \ldots, m. \end{array}$$

We can think of this problem as obtained by approximating the nonlinear function $h(u)$ by a piecewise-linear function $|u-1|+1$. As shown in the figure below, this is a good approximation around $u = 1$.



You can solve the Chebyshev approximation problem using `cvx`. The (convex) function $\|Ap - \mathbf{1}\|_\infty$ can be expressed in `cvx` as `norm(A*p-ones(n,1),inf)`. Give the resulting value of $f_0(p)$.

Exact solution. Finally, use `cvx` to solve

$$\begin{array}{ll} \text{minimize} & \max_{k=1,\ldots,n} \max(a_k^T p, 1/a_k^T p) \\ \text{subject to} & 0 \leq p_j \leq 1, \quad j = 1, \ldots, m \end{array}$$

exactly. You may find the `inv_pos()` function useful. Give the resulting (optimal) value of $f_0(p)$.

Solution: The following Matlab script finds the approximate solutions using the heuristic methods proposed, as well as the exact solution.

```
% illum_sol: finds approximate and exact solutions of
%            the illumination problem

clear all;

% load input data
illum_data;
```

```
% heuristic method 1: equal lamp powers
% ----------------------------------------
nopts=1000;
p = logspace(-3,0,nopts);
f = zeros(size(p));
for k=1:nopts
   f(k) = max(abs(log(A*p(k)*ones(m,1))));
end;
[val_equal,imin] = min(f);
p_equal = p(imin)*ones(m,1);

% heuristic method 2: least-squares with saturation
% ---------------------------------------------------
p_ls_sat = A\ones(n,1);
p_ls_sat = max(p_ls_sat,0);             % rounding negative p_i to 0
p_ls_sat = min(p_ls_sat,1);             % rounding p_i > 1 to 1
val_ls_sat = max(abs(log(A*p_ls_sat)));

% heuristic method 3: regularized least-squares
% -----------------------------------------------
rhos = linspace(1e-3,1,nopts);
crit = [];
for j=1:nopts
    p = [A; sqrt(rhos(j))*eye(m)]\[ones(n,1); sqrt(rhos(j))*0.5*ones(m,1)];
    crit = [ crit norm(p-0.5,inf) ];
end
idx = find(crit <= 0.5);
rho = rhos(idx(1));                     % smallest rho s.t. p is in [0,1]
p_ls_reg = [A; sqrt(rho)*eye(m)]\[ones(n,1); sqrt(rho)*0.5*ones(m,1)];
val_ls_reg = max(abs(log(A*p_ls_reg)));

% heuristic method 4: chebyshev approximation
% ---------------------------------------------
cvx_begin
   variable p_cheb(m)
   minimize(norm(A*p_cheb-1, inf))
   subject to
      p_cheb >= 0
      p_cheb <= 1
cvx_end
val_cheb = max(abs(log(A*p_cheb)));
```

```
% exact solution:
% ---------------
cvx_begin
   variable p_exact(m)
   minimize(max([A*p_exact; inv_pos(A*p_exact)]))
   subject to
       p_exact >= 0
       p_exact <= 1
cvx_end
val_exact = max(abs(log(A*p_exact)));

% Results
% -------
[p_equal p_ls_sat p_ls_reg p_cheb p_exact]
[val_equal val_ls_sat val_ls_reg val_cheb val_exact]
```

The results are summarized in the following table.

	method 1	method 2	method 3	method 4	exact
$f_0(p)$	0.4693	0.8628	0.4439	0.4198	0.3575
p_1	0.3448	1	0.5004	1	1
p_2	0.3448	0	0.4777	0.1165	0.2023
p_3	0.3448	1	0.0833	0	0
p_4	0.3448	0	0.0002	0	0
p_5	0.3448	0	0.4561	1	1
p_6	0.3448	1	0.4354	0	0
p_7	0.3448	0	0.4597	1	1
p_8	0.3448	1	0.4307	0.0249	0.1882
p_9	0.3448	0	0.4034	0	0
p_{10}	0.3448	1	0.4526	1	1