Linear inequalities, halfspaces, polyhedra

Exercise 1. When does one halfspace contain another? Give conditions under which

$$\{x \mid a^T x \leq b\} \subseteq \{x \mid \tilde{a}^T x \leq \tilde{b}\}$$

($a \neq 0$, $\tilde{a} \neq 0$). Also find the conditions under which the two halfspaces are equal.

Exercise 2. What is the distance between the two parallel hyperplanes $\{x \in \mathbf{R}^n \mid a^T x = b_1\}$ and $\{x \in \mathbf{R}^n \mid a^T x = b_2\}$?

Exercise 3. Which of the following sets S are polyhedra? If possible, express S in inequality form, *i.e.*, give matrices A and b such that $S = \{x \mid Ax \leq b\}$. If S is a subspace or an affine set, say so.

(a) $S = \{y_1 a_1 + y_2 a_2 \mid -1 \leq y_1 \leq 1, -1 \leq y_2 \leq 1\}$ for given $a_1, a_2 \in \mathbf{R}^n$.

(b) $S = \{x \in \mathbf{R}^n \mid x \geq 0,\ \mathbf{1}^T x = 1,\ \sum_{i=1}^n x_i a_i = b_1,\ \sum_{i=1}^n x_i a_i^2 = b_2\}$, where $a_i \in \mathbf{R}$ ($i = 1, \ldots, n$), $b_1 \in \mathbf{R}$, and $b_2 \in \mathbf{R}$ are given.

(c) $S = \{x \in \mathbf{R}^n \mid x \geq 0,\ x^T y \leq 1 \text{ for all } y \text{ with } \|y\| = 1\}$.

(d) $S = \{x \in \mathbf{R}^n \mid x \geq 0,\ x^T y \leq 1 \text{ for all } y \text{ with } \sum_i |y_i| = 1\}$.

(e) $S = \{x \in \mathbf{R}^n \mid \|x - x_0\| \leq \|x - x_1\|\}$ where $x_0, x_1 \in \mathbf{R}^n$ are given. S is the the set of points that are closer to x_0 than to x_1.

(f) $S = \{x \in \mathbf{R}^n \mid \|x - x_0\| \leq \|x - x_i\|,\ i = 1, \ldots, K\}$ where $x_0, \ldots, x_K \in \mathbf{R}^n$ are given. S is the set of points that are closer to x_0 than to the other x_i.

Exercise 4. *Linear and piecewise-linear classification.* Figure 1 shows a block diagram of a *linear classification* algorithm. The classifier has n inputs x_i. These inputs are first multiplied with coefficients a_i and added. The result $a^T x = \sum_{i=1}^n a_i x_i$ is then compared with a threshold b. If $a^T x \geq b$, the output of the classifier is $y = 1$; if $a^T x < b$, the output is $y = -1$.

Figure 2 shows the geometric interpretation of this operation. The set defined by $a^T x = b$ is a hyperplane with normal vector a. This hyperplane divides $\mathbf{R}^n$ in two halfspaces: one halfspace where $a^T x > b$, and another halfspace where $a^T x < b$. The output of the classifier in figure 1 is $y = 1$ or $y = -1$ depending on the halfspace in which x lies. If $a^T x = b$, we arbitrarily assign $+1$ to the output. The figure also shows two clusters of points that can be separated this way.

By combining linear classifiers of the form shown in figure 1, we can build classifiers that divide $\mathbf{R}^n$ in more complicated regions than halfspaces. In Figure 3 we combine four linear classifiers. The first three take the same input $x \in \mathbf{R}^2$. Their outputs y_1, y_2, and y_3 are the inputs to the fourth classifier. Make a sketch of the region of input vectors in $\mathbf{R}^2$ for which the output y is equal to 1.

Exercise 5. *Measurement with bounded errors.* A series of K measurements $y_1, \ldots, y_K \in \mathbf{R}$, are taken in order to estimate an unknown vector $x \in \mathbf{R}^q$. The measurements are related to the unknown vector x by $y_i = a_i^T x + v_i$, where v_i is a measurement noise that satisfies $|v_i| \leq \alpha$ but is otherwise unknown. The vectors a_i and the measurement noise bound α are known. Let X denote the set of vectors x that are consistent with the observations $y_1, \ldots, y_K$, *i.e.*, the set of x that could have resulted in the measured values of y_i. Show that X is a polyhedron.

Now we examine what happens when the measurements are occasionally in error, *i.e.*, for a few i we have no relation between x and y_i. More precisely suppose that I_{fault} is a subset of

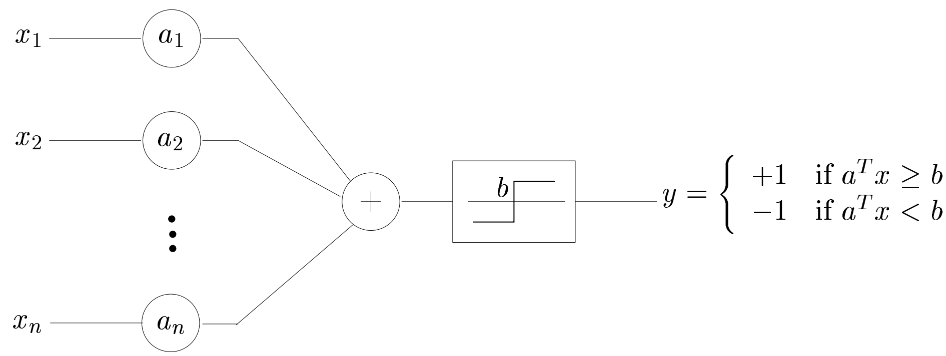


Figure 1: Block diagram of a linear classifier. The first stage computes the inner product $a^T x$ of the input x with a given vector $a = (a_1, \ldots, a_n)$. In the second stage the value of the inner product is compared with a threshold b. If $a^T x \geq b$ the output y is equal to $+1$; if $a^T x < b$, the output is -1.

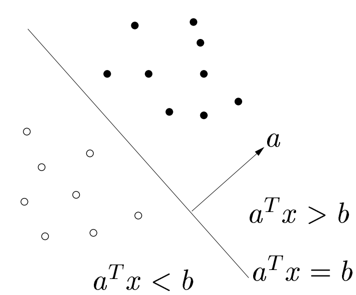


Figure 2: Geometric interpretation of linear classification.

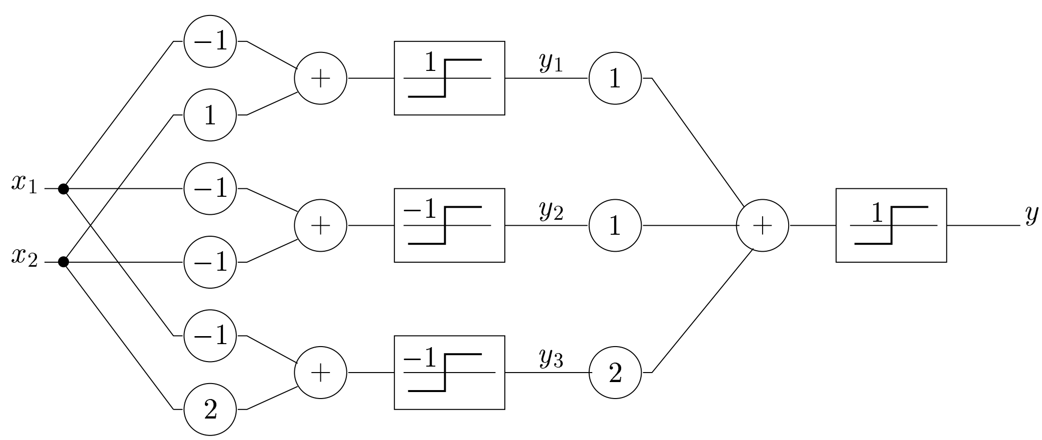


Figure 3: Block diagram of a piecewise-linear classifier.

$\{1, \ldots, K\}$, and that $y_i = a_i^T x + v_i$ with $|v_i| \le \alpha$ (as above) for $i \notin I_{\text{fault}}$, but for $i \in I_{\text{fault}}$, there is no relation between x and y_i. The set I_{fault} is the set of times of the faulty measurements.

Suppose you know that I_{fault} has at most J elements, *i.e.*, out of K measurements, at most J are faulty. You do not know I_{fault}; you know only a bound on its cardinality (size). Is X (the set of x consistent with the measurements) a polyhedron for $J > 0$?

Geometry of linear programming

Exercise 6. Consider the LP

$$\begin{array}{ll} \text{minimize} & c_1 x_1 + c_2 x_2 + c_3 x_3 \\ \text{subject to} & x_1 + x_2 \ge 1 \\ & x_1 + 2x_2 \le 3 \\ & x_1 \ge 0 \\ & x_2 \ge 0 \\ & x_3 \ge 0. \end{array}$$

Give the optimal value and the optimal set for the following values of c: $c = (-1, 0, 1)$, $c = (0, 1, 0)$, $c = (0, 0, -1)$.

Exercise 7. (a) Is $\tilde{x} = (1, 1, 1, 1)$ an extreme point of the polyhedron $\mathcal{P}$ defined by the linear inequalities

$$\begin{bmatrix} -1 & -6 & 1 & 3 \\ -1 & -2 & 7 & 1 \\ 0 & 3 & -10 & -1 \\ -6 & -11 & -2 & 12 \\ 1 & 6 & -1 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \le \begin{bmatrix} -3 \\ 5 \\ -8 \\ -7 \\ 4 \end{bmatrix} ?$$

If it is, find a vector c such that $\tilde{x}$ is the unique minimizer of $c^T x$ over $\mathcal{P}$.

(b) Same question for the polyhedron defined by the inequalities

$$\begin{bmatrix} 0 & -5 & -2 & -5 \\ -7 & -7 & -2 & -2 \\ -4 & -4 & -7 & -7 \\ -8 & -3 & -3 & -4 \\ -4 & -4 & 2 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \le \begin{bmatrix} -12 \\ -17 \\ -22 \\ -18 \\ -8 \end{bmatrix}$$

and the equality $8x_1 - 7x_2 - 10x_3 - 11x_4 = -20$.

Feel free to use Matlab (in particular the `rank` command).

Exercise 8. Consider the linear-fractional program

$$\begin{array}{ll} \text{minimize} & \dfrac{c^T x + \gamma}{d^T x + \delta} \\ \text{subject to} & Ax \le b, \end{array} \tag{1}$$

where $A \in \mathbf{R}^{m \times n}$, $b \in \mathbf{R}^m$, $c, d \in \mathbf{R}^n$, and $\gamma, \delta \in \mathbf{R}$. We assume that the polyhedron

$$\mathcal{P} = \{x \in \mathbf{R}^n \mid Ax \le b\}$$

is bounded and that $d^T x + \delta > 0$ for all $x \in \mathcal{P}$.

Show that you can solve (1) by solving the LP

$$\begin{array}{ll} \text{minimize} & c^T y + \gamma z \\ \text{subject to} & Ay - zb \le 0 \\ & d^T y + \delta z = 1 \\ & z \ge 0 \end{array} \tag{2}$$

in the variables $y \in \mathbf{R}^n$ and $z \in \mathbf{R}$. More precisely, suppose $\hat{y}$ and $\hat{z}$ are a solution of (2). Show that $\hat{z} > 0$ and that $\hat{y}/\hat{z}$ solves (1).

Some simple linear programs

Exercise 9. For each of the following LPs, give the optimal value and the optimal solution. (If the optimal solution is not unique, it is sufficient to give one optimal solution.)

(a)

$$\begin{array}{ll} \text{maximize} & c^T x \\ \text{subject to} & \mathbf{1}^T x = k \\ & 0 \le x \le \mathbf{1} \end{array}$$

with variable $x \in \mathbf{R}^n$. k is a given integer with $1 \le k \le n$.

(b)

$$\begin{array}{ll} \text{maximize} & c^T x \\ \text{subject to} & \mathbf{1}^T x \le k \\ & 0 \le x \le \mathbf{1} \end{array}$$

with variable $x \in \mathbf{R}^n$. k is a given integer with $1 \le k \le n$.

(c)

$$\begin{array}{ll} \text{maximize} & c^T x \\ \text{subject to} & -y \le x \le y \\ & \mathbf{1}^T y = k \\ & y \le \mathbf{1} \end{array}$$

with variables $x \in \mathbf{R}^n$ and $y \in \mathbf{R}^n$. k is a given integer with $1 \le k \le n$.

(d)

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T u + \mathbf{1}^T v \\ \text{subject to} & u - v = c \\ & u \ge 0, \ v \ge 0 \end{array}$$

with variables $u \in \mathbf{R}^n$ and $v \in \mathbf{R}^n$.

(e)

$$\begin{array}{ll} \text{minimize} & d_1^T u - d_2^T v \\ \text{subject to} & u - v = c \\ & u \ge 0, \ v \ge 0 \end{array}$$

with variables $u \in \mathbf{R}^n$ and $v \in \mathbf{R}^n$. We assume that $d_1 \ge d_2$.

Exercise 10. *An optimal control problem with an analytical solution.* We consider the problem of maximizing a linear function of the final state of a linear system, subject to bounds on the inputs:

$$\begin{array}{ll} \text{maximize} & d^T x(N) \\ \text{subject to} & |u(t)| \leq U, \quad t = 0, \ldots, N-1 \\ & \sum_{t=0}^{N-1} |u(t)| \leq \alpha, \end{array} \tag{3}$$

where x and u are related via the recursion

$$x(t+1) = Ax(t) + Bu(t), \quad x(0) = 0,$$

and the problem data are $d \in \mathbf{R}^n$, $U, \alpha \in \mathbf{R}$, $A \in \mathbf{R}^{n \times n}$ and $B \in \mathbf{R}^n$. The variables are the input sequence $u(0), \ldots, u(N-1)$.

(a) Express (3) as an LP.

(b) Formulate a simple algorithm for solving this LP. (It can be solved very easily, without using a general LP code.)

Hint. The problem is a variation on exercise 9, parts (a),(b),(c).

(c) Apply your method to the matrices

$$A = \begin{bmatrix} 9.9007\,10^{-1} & 9.9340\,10^{-3} & -9.4523\,10^{-3} & 9.4523\,10^{-3} \\ 9.9340\,10^{-2} & 9.0066\,10^{-1} & 9.4523\,10^{-2} & -9.4523\,10^{-2} \\ 9.9502\,10^{-2} & 4.9793\,10^{-4} & 9.9952\,10^{-1} & 4.8172\,10^{-4} \\ 4.9793\,10^{-3} & 9.5021\,10^{-2} & 4.8172\,10^{-3} & 9.9518\,10^{-1} \end{bmatrix}, \quad B = \begin{bmatrix} 9.9502\,10^{-2} \\ 4.9793\,10^{-3} \\ 4.9834\,10^{-3} \\ 1.6617\,10^{-4} \end{bmatrix}. \tag{4}$$

(You can download these data by executing the matlab file `ex10data.m` which can be found on the class webpage. The calling sequence is `[A,b] = ex10data`.)

Use

$$d = (0, 0, 1, -1), \quad N = 100, \quad U = 2, \quad \alpha = 161.$$

Plot the optimal input and the resulting sequences $x_1(t)$ and $x_2(t)$.

Remark. These data were derived as follows. We consider a system described by two second-order equations

$$\begin{aligned} m_1 \ddot{v}_1(t) &= -K(v_1(t) - v_2(t)) - D(\dot{v}_1(t) - \dot{v}_2(t)) + u(t) \\ m_2 \ddot{v}_2(t) &= K(v_1(t) - v_2(t)) + D(\dot{v}_1(t) - \dot{v}_2(t)). \end{aligned}$$

These equations describe the motion of two masses m_1 and m_2 with positions $v_1 \in \mathbf{R}$ and $v_2 \in \mathbf{R}$, respectively, and connected by a spring with spring constant K and a damper with constant D. An external force u is applied to the first mass. We use the values

$$m_1 = m_2 = 1, \quad K = 1, \quad D = 0.1,$$

so the state equations are

$$\begin{bmatrix} \ddot{v}_1(t) \\ \ddot{v}_2(t) \\ \dot{v}_1(t) \\ \dot{v}_2(t) \end{bmatrix} = \begin{bmatrix} -0.1 & 0.1 & -1.0 & 1.0 \\ 0.1 & -0.1 & 1.0 & -1.0 \\ 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \end{bmatrix} \begin{bmatrix} \dot{v}_1(t) \\ \dot{v}_2(t) \\ v_1(t) \\ v_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} u(t).$$

We discretize the system by considering inputs u that are piecewise constant with sampling interval $T = 0.1$, *i.e.*, we assume u is constant in the intervals $[0.1k, 0.1(k+1))$, for $k = 0, 1, 2, \ldots$. It can be shown that the discretized state equations are

$$z((k+1)T) = Az(kT) + Bu(kT), \quad k \in \mathbf{Z}, \tag{5}$$

where $z(t) = (\dot{v}_1(t), \dot{v}_2(t), v_1(t), v_2(t))$, and A and B given by (4).

The interpretation of the cost function $d^T x(N)$ with $d = (0, 0, 1, -1)$ is that we try to determine the input sequence that pulls the two masses as far apart as possible in N time steps.

Exercise 11. *Power allocation problem with analytical solution.* Consider a system of n transmitters and n receivers. The ith transmitter transmits with power x_i, $i = 1, \ldots, n$. The vector x is the variable in this problem. The path gain from each transmitter j to each receiver i is denoted A_{ij} and is assumed to be known. (Obviously, $A_{ij} \geq 0$, so the matrix A is elementwise nonnegative. We also assume that $A_{ii} > 0$.) The signal received by each receiver i consists of three parts: the desired signal, arriving from transmitter i with power $A_{ii}x_i$, the interfering signal, arriving from the other transmitters with power $\sum_{j \neq i} A_{ij}x_j$, and noise v_i (v_i is positive and known). We are interested in allocating the powers x_i in such a way that the signal to noise plus interference ratio (SNIR) at each of the receivers exceeds a level α. (Thus α is the minimum acceptable SNIR for the receivers; a typical value might be around $\alpha = 3$.) In other words, we want to find $x \geq 0$ such that for $i = 1, \ldots, n$

$$A_{ii}x_i \geq \alpha \left(\sum_{j \neq i} A_{ij}x_j + v_i \right).$$

Equivalently, the vector x has to satisfy the set of linear inequalities

$$x \geq 0, \quad Bx \geq \alpha v \tag{6}$$

where $B \in \mathbf{R}^{n \times n}$ is defined as

$$B_{ii} = A_{ii}, \quad B_{ij} = -\alpha A_{ij}, \ j \neq i.$$

(a) Suppose you are given a desired level of α, so the right hand side αv in (6) is a known positive vector. Show that (6) is feasible if and only if B is invertible and $z = B^{-1}\mathbf{1} \geq 0$ ($\mathbf{1}$ is the vector with all components 1). Show how to construct a feasible power allocation x from z.

(b) Show how to find the largest possible SNIR, *i.e.*, how to maximize α subject to the existence of a feasible power allocation.

Hint. You can refer to the following result from linear algebra. Let $T \in \mathbf{R}^{n \times n}$ be a matrix with nonnegative elements, and $s \in \mathbf{R}$. Then the following statements are equivalent:

(a) There exists an $x \geq 0$ with $(sI - T)x > 0$.

(b) $sI - T$ is nonsingular and the matrix $(sI - T)^{-1}$ has nonnegative elements.

(c) $s > \max_i |\lambda_i(T)|$ where $\lambda_i(T)$ ($i = 1, \ldots, n$) are the eigenvalues of T. The quantity $\rho(T) = \max_i |\lambda_i(T)|$ is called the *spectral radius* of T. It is a complicated but readily computed function of T.

(For such s, the matrix $sI - T$ is called a *nonsingular M-matrix.*)

Remark. This problem gives an analytic solution to a very special form of transmitter power allocation problem. Specifically, there are exactly as many transmitters as receivers, and no power limits on the transmitters. One consequence is that the receiver noises v_i play no role at all in the solution — just crank up all the transmitters to overpower the noises!

Formulating problems as LPs

Exercise 12. Formulate the following problems as LPs:

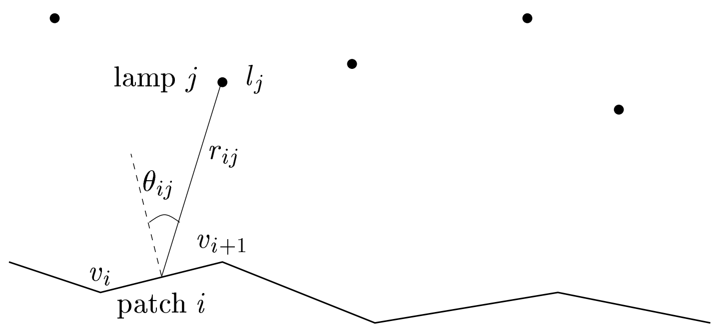
(a) minimize $\|Ax - b\|_1$ subject to $\|x\|_\infty \leq 1$.

(b) minimize $\|x\|_1$ subject to $\|Ax - b\|_\infty \leq 1$.

(c) minimize $\|Ax - b\|_1 + \|x\|_\infty$.

In each problem, $A \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$ are given, and $x \in \mathbf{R}^n$ is the optimization variable.

Exercise 13. *An illumination problem.* We consider an illumination system of m lamps, at positions $l_1, \ldots, l_m \in \mathbf{R}^2$, illuminating n flat patches. The patches are line segments; the ith patch is given by $[v_i, v_{i+1}]$ where $v_1, \ldots, v_{n+1} \in \mathbf{R}^2$. The variables in the problem are the lamp powers $p_1, \ldots, p_m$, which can vary between 0 and 1.



The illumination at (the midpoint of) patch i is denoted I_i. We will use a simple model for the illumination:

$$I_i = \sum_{j=1}^{m} a_{ij} p_j, \quad a_{ij} = r_{ij}^{-2} \max\{\cos\theta_{ij}, 0\}, \tag{7}$$

where r_{ij} denotes the distance between lamp j and the midpoint of patch i, and θ_{ij} denotes the angle between the upward normal of patch i and the vector from the midpoint of patch i to lamp j, as shown in the figure. This model takes into account "self-shading" (*i.e.*, the fact that a patch is illuminated only by lamps in the halfspace it faces) but not shading of one patch caused by another. Of course we could use a more complex illumination model, including shading and even reflections. This just changes the matrix relating the lamp powers to the patch illumination levels.

The problem is to determine lamp powers that make the illumination levels close to a given desired illumination level I_{des}, subject to the power limits $0 \leq p_i \leq 1$.

(a) Suppose we use the maximum deviation

$$\phi(p) = \max_{k=1,\ldots,n} |I_k - I_{\text{des}}|$$

as a measure for the deviation from the desired illumination level. Formulate the illumination problem using this criterion as a linear programming problem.

(b) There are several suboptimal approaches based on weighted least-squares that are often used in practice. We consider two examples.

i. *Saturated least-squares.* We can solve the least-squares problem

$$\text{minimize} \quad \sum_{k=1}^{n}(I_k - I_{\text{des}})^2$$

ignoring the constraints. If the solution is not feasible, we can saturate it, *i.e.*, set $p_j := 0$ if $p_j \le 0$ and $p_j := 1$ if $p_j \ge 1$.
Download the Matlab file `ex13data.m` from from the class webpage and generate problem data by `[A,Ides] = ex13data`. (The elements of A are the coefficients a_{ij} in (7).) Compute a feasible p using this first method, and calculate $\phi(p)$.
Note: it is very easy to solve least-squares problems in Matlab using the 'backslash' operator `\`: `x=A\b` solves the least-squares problem

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

Try '`help slash`' for details.

ii. *Weighted least-squares.* We consider another least-squares problem:

$$\text{minimize} \quad \sum_{k=1}^{n}(I_k - I_{\text{des}})^2 + \mu \sum_{i=1}^{m}(p_i - 0.5)^2,$$

where $\mu \ge 0$ is used to attach a cost to deviation of the powers from the value 0.5, which lies in the middle of the power limits. For large enough μ, the solution of this problem will satisfy $0 \le p_i \le 1$, *i.e.*, be feasible for the original problem.
Explain how you solve this problem in Matlab. For the problem data generated by `ex13data.m`, find the smallest μ such that p becomes feasible, and evaluate $\phi(p)$.

(c) Using the same data as in part (b), solve the LP you derived in part (a). Compare the solution with the solutions you obtained using the (weighted) least-squares methods of part (b).
You can solve the LP using any software you like.

- If your Matlab installation includes the optimization toolbox, you can use the function `lp.m`. Type '`help lp`' for details.
- You can download the PCx code from the class webpage. This code is fast, but we only have an executable for SUN and Windows95/98/NT. After you install the code, you can type '`help pcx`' for details on how to use it.
- You can also use the code `mpc.m`, available from the webpage. This code is entirely written in Matlab, so it runs on any machine, but it is much slower than PCx. Again, typing '`help mpc`' will provide details on how to use the code.

If you use PCx or `mpc.m`, you will notice that in order to solve an LP in inequality form

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

you have to use the calling sequence `[z,x,status] = pcx(A',-c,b)`, or `[z,x,status] = mpc(A',-c,b)`.

Exercise 14. In exercise 13, we encountered the problem

$$\begin{array}{ll} \text{minimize} & \max_{k=1,\ldots,n} |a_k^T p - I_{\text{des}}| \\ \text{subject to} & 0 \le p \le \mathbf{1} \end{array} \tag{8}$$

(with variables p). We have seen that this is readily cast as an LP.

In (8) we use the maximum of the *absolute deviations* $|I_k - I_{\rm des}|$ to measure the difference from the desired intensity. Suppose we prefer to use the *relative deviations* instead, where the relative deviation is defined as

$$\max\{I_k/I_{\rm des}, I_{\rm des}/I_k\} - 1 = \begin{cases} \dfrac{I_k - I_{\rm des}}{I_{\rm des}} & \text{if } I_k \geq I_{\rm des} \\ \dfrac{I_{\rm des} - I_k}{I_k} & \text{if } I_k \leq I_{\rm des}. \end{cases}$$

This leads us to the following formulation:

$$\begin{array}{ll} \text{minimize} & \displaystyle\max_{k=1,\ldots,n} \max\{\, a_k^T p/I_{\rm des},\ I_{\rm des}/(a_k^T p)\,\} \\ \text{subject to} & 0 \leq p \leq \mathbf{1} \\ & a_k^T p > 0, \quad k = 1, \ldots, n. \end{array} \qquad (9)$$

Explain how you can solve this using linear programming (*i.e.*, by solving one or more LPs)

Exercise 15. This exercise is concerned with a problem that has applications in VLSI. The problem is to find the optimal positions of n cells or modules on a chip. More specifically, the variables are the coordinates x_i, y_i, $i = 1, \ldots, n$, of the n cells. The cells must be placed in a square $C = \{(x, y) \mid -1 \leq x \leq 1, -1 \leq y \leq 1\}$.

Each cell has several terminals, which are connected to terminals on other cells, or to input/output (I/O) terminals on the perimeter of C. The positions of the I/O terminals are known and fixed.

The connections between the cells are specified as follows. We are given a matrix $A \in \mathbf{R}^{N \times n}$ and two vectors $b_x \in \mathbf{R}^N$, $b_y \in \mathbf{R}^N$. Each row of A and each component of b_x and b_y describe a connection between two terminals. For each $i = 1, \ldots, N$, we can distinguish two possibilities, depending on whether row i of A describes a connection between two cells, or between a cell and an I/O terminal.

- If row i describes a connection between two cells j and k (with $j < k$), then

$$a_{il} = \begin{cases} 1 & \text{if } l = j \\ -1 & \text{if } l = k \\ 0 & \text{otherwise} \end{cases}, \qquad b_{x,i} = 0, \quad b_{y,i} = 0.$$

  In other words, we have

$$a_i^T x - b_{x,i} = x_j - x_k, \quad a_i^T y - b_{y,i} = y_j - y_k.$$

- If row i describes a connection between a cell j and an I/O terminal with coordinates $(\bar{x}, \bar{y})$, then

$$a_{il} = \begin{cases} 1 & \text{if } l = j \\ 0 & \text{otherwise} \end{cases}, \qquad b_{i,x} = \bar{x}, \quad b_{i,y} = \bar{y},$$

  so we have

$$a_i^T x - b_{x,i} = x_j - \bar{x}, \quad a_i^T y - b_{y,i} = y_j - \bar{y}.$$

  (Note that the point $(\bar{x}, \bar{y})$ must lie on the boundary of C.)

Figure 4 illustrates this notation. Here, $n = 3$, $N = 6$, and A, b_x and b_y are given by

$$A = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad b_x = \begin{bmatrix} 0.0 \\ 0.0 \\ -1.0 \\ 0.5 \\ 0.0 \\ 1.0 \end{bmatrix}, \quad b_y = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \\ 1.0 \\ -1.0 \\ 0.5 \end{bmatrix}.$$

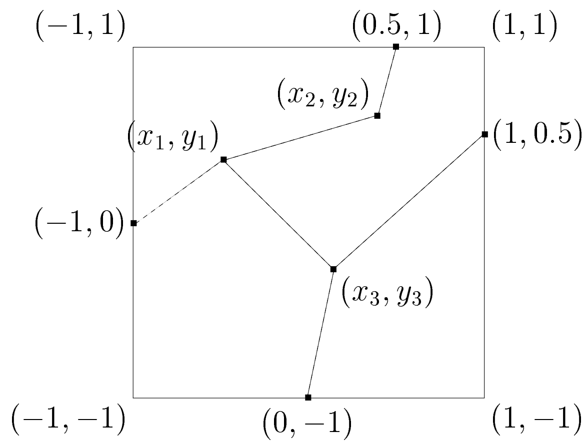


Figure 4: Example illustrating the notation in exercise 15.

The problem we consider is to determine the coordinates (x_i, y_i) that minimize some measure of the total wirelength of the connections. We can formulate different variations.

(a) Suppose we use the Euclidean distance between terminals to measure the length of a connection, and that we minimize the sum of the squares of the connection lengths. In other words we determine x and y by solving

$$\text{minimize} \sum_{i=1}^{N} \left((a_i^T x - b_{x,i})^2 + (a_i^T y - b_{y,i})^2 \right),$$

or, in matrix notation,

$$\text{minimize } \|Ax - b_x\|^2 + \|Ay - b_y\|^2. \tag{10}$$

The variables are $x \in \mathbf{R}^n$ and $y \in \mathbf{R}^n$. (Note that we don't have to add the constraints $-1 \le x_i \le 1$ and $-1 \le y_i \le 1$ explicitly, since a solution with a cell outside C can never be optimal.) Since the two terms in (10) are independent, the solution can be obtained by solving two least-squares problems, one to determine x, and one to determine y. Equivalently, we can solve two sets of linear equations

$$(A^T A)x = A^T b_x, \quad (A^T A)y = A^T b_y.$$

(b) A second and more realistic choice is to use the Manhattan distance between two connected terminals as a measure for the length of the connection, *i.e.*, to consider the optimization problem

$$\text{minimize} \sum_{i=1}^{N} \left(\left| a_i^T x - b_{x,i} \right| + \left| a_i^T y - b_{y,i} \right| \right).$$

In matrix notation, this can be written as

$$\text{minimize } \|Ax - b_x\|_1 + \|Ay - b_y\|_1.$$

(c) As a third variation, suppose we measure the length of a connection between two terminals by the Manhattan distance between the two points, as in (b) , but instead of minimizing the sum of the lengths, we minimize the maximum length, *i.e.*, we solve

$$\text{minimize } \max_{i=1,\ldots,N} \left(\left| a_i^T x - b_{x,i} \right| + \left| a_i^T y - b_{y,i} \right| \right).$$

(d) Finally, we can consider the problem

$$\text{minimize } \sum_{i=1}^{N} \left(h(a_i^T x - b_{x,i}) + h(a_i^T y - b_{y,i}) \right)$$

where h is a piecewise-linear function defined as $h(z) = \max\{z, -z, \gamma\}$ and γ is a given positive constant. The function h is plotted in figure 5.

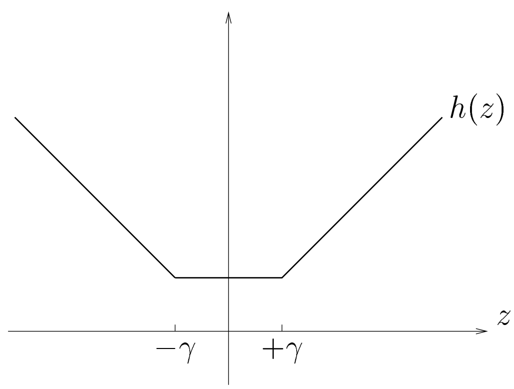


Figure 5: The function h used in problem 4.

Give LP formulations for problems (b), (c) and (d). You may introduce new variables, but you must explain clearly why your formulation and the original problem are equivalent.

Numerical example. We will also compare the solutions obtained from the four variations for a small example. For simplicity, we consider a one-dimensional version of the problem, *i.e.*, the variables are $x \in \mathbf{R}^n$, and the goal is to place the cells on the interval $[-1, 1]$. We will also drop the subscript in b_x. The four formulations of the one-dimensional placement problem are the following.

(a) ℓ^2*-placement:* minimize $\|Ax - b\|^2 = \sum_i (a_i^T x - b_i)^2$.

(b) ℓ^1*-placement:* minimize $\|Ax - b\|_1 = \sum_i |a_i^T x - b_i|$.

(c) ℓ^∞*-placement:* minimize $\|Ax - b\|_\infty = \max_i |a_i^T x - b_i|$.

(d) ℓ^1*-placement with 'dead zone':* minimize $\sum_i h(a_i^T x - b_i)$. We will use a value $\gamma = 0.02$.

To generate the data, download the file `ex15data.m` from the class webpage. The command `[A,b] = ex15data('large')` generates a problem with 100 cells and 300 connections; `[A,b] = ex15data('small')` generates a problem with with 50 cells and 150 connections. You can choose either problem. The larger problem is more interesting but may be too large if you use the student version of Matlab. It will also take much longer to solve if you don't use the PCx LP solver (see below).

A few remarks and suggestions:

- The preferred way in Matlab to solve a least squares problem

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

  is to use the 'backslash' operator, `x = A\b`. An alternative is `x = (A'*A)\(A'*b)`, which is somewhat faster but less accurate.

- The other three problems require an LP solver. There are two possibilities. The first is to download the PCx code from the class webpage. This code exploits sparsity and is quite fast, but unfortunately we only have an executable for SUN and Windows95/98/NT. The second possibility is to use the code `mpc.m`, also available from the class webpage. This code is entirely written in Matlab, so it runs on any machine, but it is much slower than PCx.

  After you install the code, you can type '`help pcx`' or '`help mpc`' for more details on how to use the codes. You will notice that in order to solve an LP in inequality form

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

  you have to use the calling sequence `[z,x,status] = pcx(A',-c,b)`, or `[z,x,status] = mpc(A',-c,b)`.

  If you use `mpc`, make sure that the data type of the first argument is 'sparse'. The code will run much slower if the argument is a full matrix (type `help sparse` for details).

Compare the solutions obtained by the four methods.

- Plot a histogram of the n positions x_i for each solution (using the `hist` command). To make the comparison easier, it is a good idea to put the four histograms on one figure using `subplot`.
- Also plot a histogram of the connnection lengths $|a_i^T x - b_i|$.
- Compute the total wire length $\sum_i |a_i^T x - b_i|$ for each of the four solutions.
- Compute the length of the longest connection $\max_i |a_i^T x - b_i|$ for each of the four solutions.
- So far we have assumed that the cells have zero width. In practice we have to take overlap between cells into account. Assume that two cells i and j overlap when $|x_i - x_j| \leq 0.01$. For each of the four solutions, calculate how many pairs of cells overlap. You can express the overlap as a percentage of the total number $n(n-1)/2$ of pairs of cells.

Are the results what you expect? Which of the four solutions would you prefer if the most important criteria are total wirelength $\sum_i |a_i^T x - b_i|$ and overlap?

Exercise 16. Suppose you are given two sets of points $\{v^1, v^2, \ldots, v^K\}$ and $\{w^1, w^2, \ldots, w^L\}$ in $\mathbf{R}^n$. Can you formulate the following two problems as LP feasibility problems?

(a) Determine a hyperplane that separates the two sets, *i.e.*, find $a \in \mathbf{R}^n$ and $b \in \mathbf{R}$ with $a \neq 0$ such that

$$a^T v^i \leq b, \quad i = 1, \ldots, K,$$
$$a^T w^i \geq b, \quad i = 1, \ldots, L.$$

Note that we require $a \neq 0$, so you have to make sure your method does not return the trivial solution $a = 0$, $b = 0$. Figure 6 shows an example.

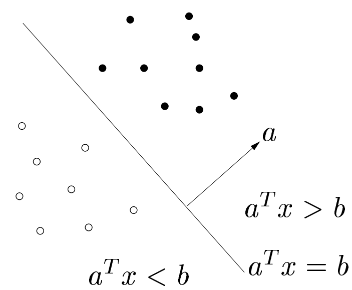


Figure 6: A hyperplane separating two sets of points.

(b) Determine a sphere separating the two sets of points, *i.e.*, find $x_c \in \mathbf{R}^n$ and $R \geq 0$ such that

$$(v^i - x_c)^T (v^i - x_c) \leq R^2, \quad i = 1, \ldots, K,$$
$$(w^i - x_c)^T (w^i - x_c) \geq R^2, \quad i = 1, \ldots, L.$$

(x_c is the center of the sphere; R is its radius.)

Exercise 17. *Robot grasp problem with static friction.*

We consider a rigid object held by N robot fingers. For simplicity we assume that the object and all forces acting on it lie in a plane (figure 7). The fingers make contact with the object

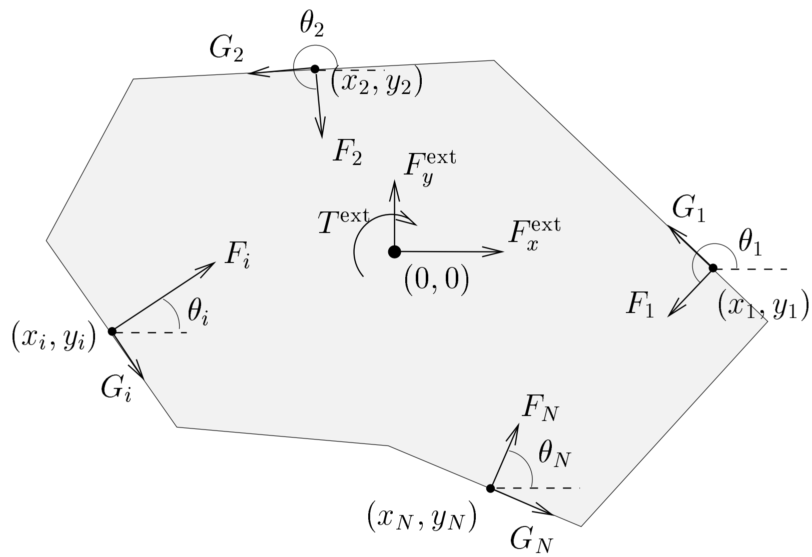


Figure 7: A rigid object held by N robot fingers.

at points (x_i, y_i), $i = 1, \ldots, N$. (Although it does not matter, you can assume that the origin $(0, 0)$ is at the center of gravity of the object.) Each finger applies a force with magnitude

F_i on the object, in a direction normal to the surface at that contact point, and pointing towards the object. The horizontal component of the ith contact force is equal to $F_i \cos\theta_i$, and the vertical component is $F_i \sin\theta_i$, where θ_i is the angle between the inward pointing normal to the surface and a horizontal line.

At each contact point there is a friction force G_i which is tangential to the surface. The horizontal component is $G_i \sin\theta_i$ and the vertical component is $-G_i \cos\theta_i$. The orientation of the friction force is arbitrary (*i.e.*, G_i can be positive or negative), but its magnitude $|G_i|$ cannot exceed μF_i, where $\mu \geq 0$ is a given constant (the friction coefficient).

Finally, there are several external forces and torques that act on the object. We can replace those external forces by equivalent horizontal and vertical forces $F_x^{\rm ext}$ and $F_y^{\rm ext}$ at the origin, and an equivalent torque $T^{\rm ext}$. These two external forces and the external torque are given.

The static equilibrium of the object is characterized by the following three equations:

$$\sum_{i=1}^{N}(F_i \cos\theta_i + G_i \sin\theta_i) + F_x^{\rm ext} = 0 \tag{11}$$

(the horizontal forces add up to zero),

$$\sum_{i=1}^{N}(F_i \sin\theta_i - G_i \cos\theta_i) + F_y^{\rm ext} = 0 \tag{12}$$

(the vertical forces add up to zero),

$$\sum_{i=1}^{N}((F_i \cos\theta_i + G_i \sin\theta_i)y_i - (F_i \sin\theta_i - G_i \cos\theta_i)x_i) + T^{\rm ext} = 0 \tag{13}$$

(the total torque is zero). As mentioned above, we assume the friction model can be expressed as a set of inequalities

$$|G_i| \leq \mu F_i, \quad i = 1, \ldots, N. \tag{14}$$

If we had no friction, then $N = 3$ fingers would in general be sufficient to hold the object, and we could find the forces F_i by solving the three linear equations (11)-(13) for the variables F_i. In there is friction, or $N > 3$, we have more unkown forces than equilibrium equations, so the system of equations is underdetermined. We can then take advantage of the additional degrees of freedom to find a set of forces F_i that are 'small'. Express the following two problems as LPs.

(a) Find the set of forces F_i that minimizes $\sum_{i=1}^{N} F_i$ subject to the constraint that the object is in equilibrium. More precisely, the constraint is that there exist friction forces G_i that, together with F_i, satisfy (11)-(14).

(b) Find a set of forces F_i that minimizes $\max_{i=1,\ldots,N} F_i$ subject to the constraint that the object is in equilibrium.

Which of these two problems do you expect will have a solution with a larger number of F_i's equal to zero?

Exercise 18. *Robust linear programming.*

(a) Let $x \in \mathbf{R}^n$ be a given vector. Prove that $x^T y \leq \|x\|_1$ for all y with $\|y\|_\infty \leq 1$. Is the inequality tight, *i.e.*, does there exist a y that satisfies $\|y\|_\infty \leq 1$ and $x^T y = \|x\|_1$?

(b) Consider the set of linear inequalities

$$a_i^T x \leq b_i, \quad i = 1, \ldots, m. \tag{15}$$

Suppose you don't know the coefficients a_i exactly. Instead you are given nominal values $\bar{a}_i$, and you know that the actual coefficient vectors satisfy

$$\|a_i - \bar{a}_i\|_\infty \leq \rho$$

for a given $\rho > 0$. In other words the actual coefficients a_{ij} can be anywhere in the intervals $[\bar{a}_{ij} - \rho, \bar{a}_{ij} + \rho]$, or equivalently, each vector a_i can lie anywhere in a rectangle with corners $\bar{a}_i + v$ where $v \in \{-\rho, \rho\}^n$ (*i.e.*, v has components ρ or $-\rho$).

The set of inequalities (15) must be satisfied for all possible values of a_i, *i.e.*, we replace (15) with the constraints

$$a_i^T x \leq b_i \text{ for all } a_i \in \{\bar{a}_i + v \mid \|v\|_\infty \leq \rho\} \text{ and for } i = 1, \ldots, m. \tag{16}$$

A straightforward but very inefficient way to express this constraint is to enumerate the 2^n corners of the rectangle of possible values a_i and to require that

$$\bar{a}_i^T x + v^T x \leq b_i \text{ for all } v \in \{-\rho, \rho\}^n \text{ and for } i = 1, \ldots, m.$$

This is a system of $m2^n$ inequalities.

Use the result in (a) to show that (16) is in fact equivalent to the much more compact set of nonlinear inequalities

$$\bar{a}_i^T x + \rho\|x\|_1 \leq b_i, \quad i = 1, \ldots, m. \tag{17}$$

(c) Consider the LP

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b_i, \quad i = 1, \ldots, m. \end{array}$$

Again we are interested in situations where the coefficient vectors a_i are uncertain, but satisfy bounds $\|a_i - \bar{a}_i\|_\infty \leq \rho$ for given $\bar{a}_i$ and ρ. We want to minimize $c^T x$ subject to the constraint that the inequalities $a_i^T x \leq b_i$ are satisfied for *all* possible values of a_i. We call this a *robust LP*:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b_i \text{ for all } a_i \in \{\bar{a}_i + v \mid \|v\|_\infty \leq \rho\} \text{ and for } i = 1, \ldots, m. \end{array} \tag{18}$$

It follows from (b) that we can express this problem as a nonlinear optimization problem

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \bar{a}_i^T x + \rho\|x\|_1 \leq b_i, \quad i = 1, \ldots, m. \end{array} \tag{19}$$

Express (19) as an LP.

Solving (19) is a *worst-case* approach to dealing with uncertainty in the data. If $x^\star$ is the optimal solution of (19), then for any *specific* value of a_i, it may be possible to find feasible x with a lower objective value than $x^\star$. However such an x would be infeasible for some other value of a_i.

Exercise 19. *Robust Chebychev approximation.*
In a similar way, we can consider Chebychev approximation problems

$$\text{minimize } \|Ax - b\|_\infty$$

in which $A \in \mathbf{R}^{m\times n}$ is uncertain. Suppose we can characterize the uncertainty as follows. The values of A depend on parameters $u \in \mathbf{R}^p$, which are unknown but satisfy $\|u\|_\infty \le \rho$. Each row vector a_i can be written as $a_i = \bar{a}_i + B_i u$ where $\bar{a}_i \in \mathbf{R}^n$ and $B_i \in \mathbf{R}^{n\times p}$ are given. In the *robust Chebychev approximation* we minimize the worst-case value of $\|Ax-b\|_\infty$. This problem can be written as

$$\text{minimize } \max_{\|u\|_\infty \le \rho} \max_{i=1,\ldots,m} \left|(\bar{a}_i + B_i u)^T x - b_i\right|. \tag{20}$$

Show that (20) is equivalent to

$$\text{minimize } \max_{i=1,\ldots,m} \left(|\bar{a}_i^T x - b_i| + \rho\|B_i^T x\|_1\right). \tag{21}$$

To prove this you can use the results from exercise 18. There is also a fairly straightforward direct proof. Express (21) as an LP.

Exercise 20. Describe how you would use linear programming to solve the following problem. You are given an LP

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \le b \end{array} \tag{22}$$

in which the coefficients of $A \in \mathbf{R}^{m\times n}$ are uncertain. Each coefficient A_{ij} can take arbitrary values in the interval

$$[\bar{A}_{ij} - \Delta A_{ij}, \bar{A}_{ij} + \Delta A_{ij}],$$

where $\bar{A}_{ij}$ and ΔA_{ij} are given with $\Delta A_{ij} \ge 0$. The optimization variable x in (22) must be feasible for all possible values of A. In other words, we want to solve

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \le b \text{ for all } A \in \mathcal{A} \end{array}$$

where $\mathcal{A} \subseteq \mathbf{R}^{m\times n}$ is the set

$$\mathcal{A} = \{A \in \mathbf{R}^{m\times n} \mid \bar{A}_{ij} - \Delta A_{ij} \le A_{ij} \le \bar{A}_{ij} + \Delta A_{ij},\ i = 1,\ldots,m,\ j = 1,\ldots,n\}.$$

If you know more than one solution method, you should give the most efficient one.

Exercise 21. *Feedback design for a static linear system.* In this problem we use linear programming to design a linear feedback controller for a static linear system. (The method extends to dynamical systems but we will not consider the extension here.) Figure 8 shows the system and the controller.

The elements of the vector $w \in \mathbf{R}^{n_w}$ are called the *exogeneous inputs*, $z \in \mathbf{R}^{n_z}$ are the *critical outputs*, $y \in \mathbf{R}^{n_y}$ are the *sensed outputs*, and $u \in \mathbf{R}^{n_u}$ are the *actuator inputs*. These vectors are related as

$$\begin{aligned} z &= P_{zw} w + P_{zu} u \\ y &= P_{yw} w + P_{yu} u, \end{aligned} \tag{23}$$

where the matrices P_{zu}, P_{zw}, P_{yu}, P_{yw} are given.

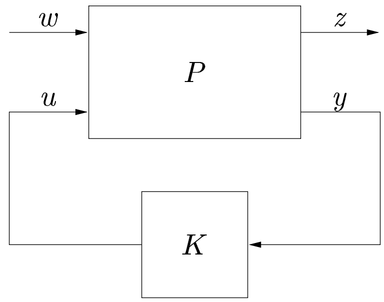


Figure 8: Closed loop control system.

The controller feeds back the sensed outputs y to the actuator inputs u. The relation is

$$u = Ky \tag{24}$$

where $K \in \mathbf{R}^{n_u \times n_y}$. The matrix K will be the design variable.

Assuming $I - P_{yu}$ is invertible, we can eliminate y from the second equation in (23). We have

$$y = (I - P_{yu}K)^{-1}P_{yw}w$$

and substituting in the first equation we can write $z = Hw$ with

$$H = P_{zw} + P_{zu}K(I - P_{yu}K)^{-1}P_{yw}. \tag{25}$$

The matrix H is a complicated nonlinear function of K.

Suppose that the signals w are disturbances or noises acting on the system, and that they can take any values with $\|w\|_\infty \le \rho$ for some given ρ. We would like to choose K so that the effect of the disturbances w on the output z is minimized, *i.e.*, we would like z to be as close as possible to zero, regardless of the values of w. Specifically, if we use the infinity norm $\|z\|_\infty$ to measure the size of z, we are interested in determining K by solving the optimization problem

$$\text{minimize} \max_{\|w\|_\infty \le \rho} \|Hw\|_\infty, \tag{26}$$

where H depends on the variable K through the formula (25).

(a) We first derive an explicit expression for the objective function in (26). Show that

$$\max_{\|w\|_\infty \le \rho} \|Hw\|_\infty = \rho \max_{i=1,\ldots,n_z} \sum_{j=1,\ldots,n_w} |h_{ij}|$$

where h_{ij} are the elements of H. Up to the constant ρ, this is the maximum row sum of H: for each row of H we calculate the sum of the absolute values of its elements; we then select the largest of these row sums.

(b) Using this expression, we can reformulate problem (26) as

$$\text{minimize } \rho \max_{i=1,\ldots,n_z} \sum_{j=1,\ldots,n_w} |h_{ij}|, \tag{27}$$

where h_{ij} depends on the variable K through the formula (25). Formulate (26) as an LP.

Hint. Use a change of variables

$$Q = K(I - P_{yu}K)^{-1},$$

and optimize over $Q \in \mathbf{R}^{n_u \times n_y}$ instead of K. You may assume that $I + QP_{yu}$ is invertible, so the transformation is invertible: we can find K from Q as $K = (I + QP_{yu})^{-1}Q$.

Exercise 22. Formulate the following problem as an LP. Find the largest ball

$$\mathcal{B}(x_c, R) = \{ \, x \mid \|x - x_c\| \leq R \, \}$$

enclosed in a given polyhedron

$$\mathcal{P} = \{ \, x \mid a_i^T x \leq b_i, \;\; i = 1, \ldots, m \, \}.$$

In other words, express the problem

$$\begin{array}{ll} \text{maximize} & R \\ \text{subject to} & \mathcal{B}(x_c, R) \subseteq \mathcal{P} \end{array}$$

as an LP. The problem variables are the center $x_c \in \mathbf{R}^n$ and the radius R of the ball.

Exercise 23. Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two polyhedra described as

$$\mathcal{P}_1 = \{x \mid Ax \leq b\}, \quad \mathcal{P}_2 = \{x \mid -\mathbf{1} \leq Cx \leq \mathbf{1}\},$$

where $A \in \mathbf{R}^{m \times n}$, $C \in \mathbf{R}^{p \times n}$, and $b \in \mathbf{R}^m$. The polyhedron $\mathcal{P}_2$ is symmetric about the origin, *i.e.*, if $x \in \mathcal{P}_2$, then $-x \in \mathcal{P}_2$. We say the origin is the *center* of $\mathcal{P}_2$.

For $t > 0$ and $x_c \in \mathbf{R}^n$, we use the notation $t\mathcal{P}_2 + x_c$ to denote the polyhedron

$$t\mathcal{P}_2 + x_c = \{tx + x_c \mid x \in \mathcal{P}_2\},$$

which is obtained by first scaling $\mathcal{P}_2$ by a factor t about the origin, and then translating its center to x_c.

Explain how you would solve the following two problems using linear programming. If you know different formulations, you should choose the most efficient method.

(a) Find the largest polyhedron $t\mathcal{P}_2 + x_c$ enclosed in $\mathcal{P}_1$, *i.e.*,

$$\begin{array}{ll} \text{maximize} & t \\ \text{subject to} & t\mathcal{P}_2 + x_c \subseteq \mathcal{P}_1. \end{array}$$

(b) Find the smallest polyhedron $t\mathcal{P}_2 + x_c$ containing $\mathcal{P}_1$, *i.e.*,

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \mathcal{P}_1 \subseteq t\mathcal{P}_2 + x_c. \end{array}$$

In both problems the variables are $t \in \mathbf{R}$ and $x_c \in \mathbf{R}^n$.

Exercise 24. Consider the linear system of exercise 10, equation (5). We study two optimal control problems. In both problems we assume the system is initially at rest at the origin, *i.e.*, $z(0) = 0$.

(a) In the first problem we want to determine the most efficient input sequence $u(kT)$, $k = 0, \ldots, 79$, that brings the system to state $(0, 0, 10, 10)$ in 80 time periods (*i.e.*, at $t = 8$ the two masses should be at rest at position $v_1 = v_2 = 10$). We assume the cost (*e.g.*, fuel consumption) of the input signal u is proportional to $\sum_k |u(kT)|$. We also impose the constraint that the amplitude of the input must not exceed 2. This leads us to the following problem:

$$\begin{array}{ll} \text{minimize} & \sum_{k=0}^{79} |u(kT)| \\ \text{subject to} & z(80T) = (0, 0, 10, 10) \\ & |u(kT)| \leq 2, \quad k = 0, \ldots, 79. \end{array} \tag{28}$$

The state z and the input u are related by (5) with $z(0) = 0$. The variables in (28) are $u(0)$, $u(T)$, $\ldots$, $u(79T)$.

(b) In the second problem we want to bring the system to the state $(0, 0, 10, 10)$ as quickly as possible, subject to the limit on the magnitude of u:

$$\begin{array}{ll} \text{minimize} & N \\ \text{subject to} & z(NT) = (0, 0, 10, 10) \\ & |u(kT)| \leq 2, \quad k = 0, \ldots, N-1. \end{array}$$

The variables are $N \in \mathbf{Z}$, and $u(0)$, $u(T)$, $\ldots$, $u(N-1)T$.

Solve these two problems numerically. Plot the input u and the positions v_1, v_2 as a function of time.

Remarks.

- These problems involve LPs with equality and inequality constraints. As we already noted before the code `pcx.m` does not directly accept an LP in such a format. There are two ways to solve an LP of the form

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \leq b \\ & Cx = d \end{array} \tag{29}$$

using `pcx.m`. The first option is to write the problem as an LP in inequality form

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

As we saw in exercise 10, we can solve this LP by calling

```
[z,x,s] = pcx(Atilde, btilde, ctilde)
```

where the input arguments are defined as

$$\tilde{A} = \begin{bmatrix} A \\ C \\ -C \end{bmatrix}^T, \quad \tilde{b} = -c, \quad \tilde{c} = \begin{bmatrix} b \\ d \\ -d \end{bmatrix}.$$

The second possibility is to convert the LP to standard form as described on page 4-5 of the lecture notes. We introduce a slack variable s and write the variable x as a difference of two nonnegative vectors x^+ and x^-:

$$\begin{array}{ll} \text{minimize} & c^T x^+ - c^T x^- \\ \text{subject to} & Ax^+ - Ax^- + s = b \\ & Cx^+ - Cx^- = d \\ & x^+ \geq 0, \; x^- \geq 0, \; s \geq 0. \end{array}$$

We can solve this LP using

```
[xhat,zhat,s] = pcx(Ahat, bhat, chat)
```

where the input arguments are defined as

$$\hat{A} = \begin{bmatrix} A & -A & I \\ C & -C & 0 \end{bmatrix}, \quad \hat{b} = \begin{bmatrix} b \\ d \end{bmatrix}, \quad \hat{c} = \begin{bmatrix} c \\ -c \\ 0 \end{bmatrix}.$$

and the variable $\hat{x}$ is defined as

$$\hat{x} = \begin{bmatrix} x^+ \\ x^- \\ s \end{bmatrix}.$$

It is useful to keep both possibilities in mind. Depending on the application, one method might lead to a smaller LP than the other.

- If you use `mpc.m` you should be aware that the code does not attempt to detect infeasibility. When applied to an infeasible problem, the code will simply stop when a maximum number of iterations is reached, while giving warnings such as 'matrix is nearly singular or badly scaled'.

Exercise 25. *Approximating a matrix in infinity norm.* The infinity (induced) norm of a matrix $A \in \mathbf{R}^{m \times n}$, denoted $\|A\|_{\infty,i}$, is defined as

$$\|A\|_{\infty,i} = \max_{i=1,\ldots,m} \sum_{j=1}^{n} |a_{ij}|.$$

The infinity norm gives the maximium ratio of the infinity norm of Ax to the infinity norm of x:

$$\|A\|_{\infty,i} = \max_{x \neq 0} \frac{\|Ax\|_\infty}{\|x\|_\infty}.$$

This norm is sometimes called the max-row-sum norm, for obvious reasons.

Consider the problem of approximating a matrix, in the max-row-sum norm, by a linear combination of other matrices. That is, we are given $k+1$ matrices $A_0, \ldots, A_k \in \mathbf{R}^{m \times n}$, and need to find $x \in \mathbf{R}^k$ that minimizes

$$\|A_0 + x_1 A_1 + \cdots + x_k A_k\|_{\infty,i}.$$

Express this problem as a linear program. Explain the significance of any extra variables in your LP. Carefully explain why your LP formulation solves this problem, *e.g.*, what is the relation between the feasible set for your LP and this problem?

Exercise 26. We are given p matrices $A_i \in \mathbf{R}^{n\times n}$, and we would like to find a single matrix $X \in \mathbf{R}^{n\times n}$ that we can use as an approximate right-inverse for each matrix A_i, *i.e.*, we would like to have

$$A_i X \approx I, \quad i = 1, \ldots, p.$$

We can do this by solving the following optimization problem with X as variable:

$$\text{minimize} \ \max_{i=1,\ldots,p} \|I - A_i X\|_\infty. \tag{30}$$

Here $\|H\|_\infty$ is the 'infinity-norm' or 'max-row-sum norm' of a matrix H, defined as

$$\|H\|_\infty = \max_{i=1,\ldots,m} \sum_{j=1}^{n} |H_{ij}|,$$

if $H \in \mathbf{R}^{m\times n}$.

Express the problem (30) as an LP. You don't have to reduce the LP to a canonical form, as long as you are clear about what the variables are, what the meaning is of any auxiliary variables that you introduce, and why the LP is equivalent to the problem (30).

Exercise 27. Explain how you would use linear programming to solve the following optimization problems.

(a) Given $A \in \mathbf{R}^{m\times n}$, $b \in \mathbf{R}^m$,

$$\text{minimize} \ \sum_{i=1}^{m} \max\{0, a_i^T x + b_i\}.$$

The variable is $x \in \mathbf{R}^n$.

(b) Given $A \in \mathbf{R}^{m\times n}$, $b \in \mathbf{R}^m$,

$$\begin{array}{ll} \text{minimize} & \max_{i=1,\ldots,m} \max\{a_i^T x + b_i, 1/(a_i^T x + b_i)\} \\ \text{subject to} & Ax + b > 0. \end{array}$$

The variable is $x \in \mathbf{R}^n$.

(c) Given m numbers $a_1, a_2, \ldots, a_m \in \mathbf{R}$, and two vectors $l, u \in \mathbf{R}^m$, find the polynomial $f(t) = c_0 + c_1 t + \cdots + c_n t^n$ of lowest degree that satisfies the bounds

$$l_i \leq f(a_i) \leq u_i, \ \ i = 1, \ldots, m.$$

The variables in the problem are the coefficients c_i of the polynomial.

(d) Given $p+1$ matrices $A_0, A_1, \ldots, A_p \in \mathbf{R}^{m\times n}$, find the vector $x \in \mathbf{R}^p$ that minimizes

$$\max_{\|y\|_1=1} \|(A_0 + x_1 A_1 + \cdots + x_p A_p)y\|_1 .$$

Exercise 28. Suppose you are given an infeasible set of linear inequalities

$$a_i^T x \leq b_i, \quad i = 1, \ldots, m,$$

and you are asked to find an x that satisfies many of the inequalities (ideally, as many as possible). Of course, the exact solution of this problem is difficult and requires combinatorial or integer optimization techniques, so you should concentrate on heuristic or sub-optimal methods. More specifically, you are asked to formulate a heuristic method based on solving a single LP.

Test the method on the example problem in the m-file `ex28data.m` available on the class webpage. (The Matlab command `[A,b] = ex28data` generates a sparse matrix $A \in \mathbf{R}^{100\times 50}$ and a vector $b \in \mathbf{R}^{100}$, that define an infeasible set of linear inequalities.) To count the number of inequalities satisfied by x, you can use the Matlab command

```
length(find(b-A*x > -1e-5)).
```

Duality

Exercise 29. We have seen in lecture 8 that the two LPs

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

and

$$\begin{array}{ll} \text{maximize} & -b^T z \\ \text{subject to} & A^T z + c = 0 \\ & z \geq 0 \end{array}$$

have equal optimal value, except when they are both infeasible. Give an example in which both problems are infeasible.

Exercise 30. Consider the LP

$$\begin{array}{ll} \text{minimize} & 47x_1 + 93x_2 + 17x_3 - 93x_4 \\ \text{subject to} & \left[\begin{array}{rrrr} -1 & -6 & 1 & 3 \\ -1 & -2 & 7 & 1 \\ 0 & 3 & -10 & -1 \\ -6 & -11 & -2 & 12 \\ 1 & 6 & -1 & -3 \end{array}\right] \left[\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array}\right] \leq \left[\begin{array}{r} -3 \\ 5 \\ -8 \\ -7 \\ 4 \end{array}\right] \end{array}$$

(*i.e.*, the same A and b as in exercise 7).

Prove, without using any LP code, that $x = (1,1,1,1)$ is optimal.

Hint. Prove that x is primal feasible, and construct a dual feasible z that shows that x is optimal. To construct such a z, follow the argument on pages 8-16 and 8-17 of the lecture notes.

Exercise 31. Consider the polyhedron

$$\mathcal{P} = \{x \in \mathbf{R}^4 \mid Ax \leq b, \;\; Cx = d\}$$

where

$$A = \left[\begin{array}{rrrr} -1 & -1 & -3 & -4 \\ -4 & -2 & -2 & -9 \\ -8 & -2 & 0 & -5 \\ 0 & -6 & -7 & -4 \end{array}\right], \quad b = \left[\begin{array}{r} -8 \\ -17 \\ -15 \\ -17 \end{array}\right]$$

and

$$C = \left[\begin{array}{cccc} 13 & 11 & 12 & 22 \end{array}\right], \quad d = 58.$$

(a) Prove that $\hat{x} = (1, 1, 1, 1)$ is an extreme point of $\mathcal{P}$.

(b) Prove that $\hat{x}$ is optimal for the LP

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

with $c = (59, 39, 38, 85)$.

(c) Is $\hat{x}$ the only optimal point? If not, describe the entire optimal set.

You can use any software, but you have to justify your answers analytically. For example, if you use an LP code, you answers must be convincing to someone who does not believe your LP code works correctly.

Exercise 32. Consider the LP

$$\begin{array}{ll} \text{minimize} & 47x_1 + 93x_2 + 17x_3 - 93x_4 \\ \text{subject to} & \begin{bmatrix} -1 & -6 & 1 & 3 \\ -1 & -2 & 7 & 1 \\ 0 & 3 & -10 & -1 \\ -6 & -11 & -2 & 12 \\ 1 & 6 & -1 & -3 \\ 11 & 1 & -1 & -8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \leq \begin{bmatrix} -3 \\ 5 \\ -8 \\ -7 \\ 4 \\ 5 \end{bmatrix} + \epsilon \begin{bmatrix} 1 \\ -3 \\ 13 \\ 46 \\ -2 \\ -75 \end{bmatrix}. \end{array} \tag{31}$$

where $\epsilon \in \mathbf{R}$ is a parameter. For $\epsilon = 0$, this is the LP of exercise 30, with one extra inequality (the sixth inequality). This inequality is inactive at $\hat{x} = (1, 1, 1, 1)$, so $\hat{x}$ is also the optimal solution for (31) when $\epsilon = 0$.

(a) Determine the range of values of ϵ for which the first four constraints are active at the optimum.

(b) Give an explicit expression for the optimal primal solution, the optimal dual solution, and the optimal value, within the range of ϵ you determined in part (a). (If for some value of ϵ the optimal points are not unique, it is sufficient to give one optimal point.)

Exercise 33. Consider the parametrized primal and dual LPs

$$\begin{array}{ll} \text{minimize} & (c + \epsilon d)^T x \\ \text{subject to} & Ax \leq b \end{array}$$

and

$$\begin{array}{ll} \text{maximize} & -b^T z \\ \text{subject to} & A^T z + c + \epsilon d = 0 \\ & z \geq 0 \end{array}$$

where

$$A = \begin{bmatrix} 2 & 3 & 5 & -4 \\ 2 & -1 & -3 & 4 \\ -2 & -1 & 3 & 1 \\ -4 & 2 & 4 & -2 \\ 2 & -3 & -9 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 6 \\ 2 \\ 1 \\ 0 \\ -8 \end{bmatrix},$$

$$c = (8, -32, -66, 14), \quad d = (-16, -6, -2, 3).$$

(a) Prove that $x^\star = (1,1,1,1)$ and $z^\star = (9,9,4,9,0)$ are optimal when $\epsilon = 0$.

(b) How does $p^\star(\epsilon)$ vary as a function of ϵ around $\epsilon = 0$? Give an explicit expression for $p^\star(\epsilon)$, and specify the interval in which it is valid.

(c) Also give an explicit expression for the primal and dual optimal solutions for values of ϵ around $\epsilon = 0$.

Remark: The problem is similar to the sensitivity problem discussed on pages 9-6 and 9-7 of the lecture notes. Here we consider the case where c is subject to a perturbation, while b is fixed, so you have to develop the 'dual' of the derivation in the lecture notes.

Exercise 34. Consider the pair of primal and dual LPs

$$\begin{array}{ll} \text{minimize} & (c+\epsilon d)^T x \\ \text{subject to} & Ax \le b + \epsilon f \end{array}$$

and

$$\begin{array}{ll} \text{maximize} & -(b+\epsilon f)^T z \\ \text{subject to} & A^T z + c + \epsilon d = 0 \\ & z \ge 0 \end{array}$$

where

$$A = \begin{bmatrix} -4 & 12 & -2 & 1 \\ -17 & 12 & 7 & 11 \\ 1 & 0 & -6 & 1 \\ 3 & 3 & 22 & -1 \\ -11 & 2 & -1 & -8 \end{bmatrix}, \quad b = \begin{bmatrix} 8 \\ 13 \\ -4 \\ 27 \\ -18 \end{bmatrix}, \quad c = \begin{bmatrix} 49 \\ -34 \\ -50 \\ -5 \end{bmatrix}, \quad d = \begin{bmatrix} 3 \\ 8 \\ 21 \\ 25 \end{bmatrix}, \quad f = \begin{bmatrix} 6 \\ 15 \\ -13 \\ 48 \\ 8 \end{bmatrix}$$

and ϵ is a parameter. Answer the following questions without using any LP software. (Of course you can check your answers numerically, but you have to justify everything independently.)

(a) Prove that $x^\star = (1,1,1,1)$ is optimal when $\epsilon = 0$, by constructing a dual optimal point $z^\star$ that has the same objective value as $x^\star$. Are there any other primal or dual optimal solutions?

(b) Express the optimal value $p^\star(\epsilon)$ as a continuous function of ϵ on an interval that contains $\epsilon = 0$. Specify the interval in which your expression is valid. Also give explicit expressions for the primal and dual solutions as a function of ϵ over the same interval.

Exercise 35. Consider the following optimization problem in x:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \|Ax+b\|_1 \le 1 \end{array} \tag{32}$$

where $A \in \mathbf{R}^{m\times n}$, $b \in \mathbf{R}^m$, $c \in \mathbf{R}^n$.

(a) Formulate this problem as a an LP in inequality form and explain why your LP formulation is equivalent to problem (32).

(b) Derive the dual LP, and show that it is equivalent to the problem

$$\begin{array}{ll} \text{maximize} & b^T z - \|z\|_\infty \\ \text{subject to} & A^T z + c = 0. \end{array}$$

What is the relation between the optimal z and the optimal variables in the dual LP?

(c) Give a direct argument (*i.e.*, not quoting any results from LP duality) that whenever x is primal feasible (*i.e.*, $\|Ax+b\|_1 \leq 1$) and z is dual feasible (*i.e.*, $A^Tz+c=0$), we have

$$c^Tx \geq b^Tz - \|z\|_\infty.$$

Exercise 36. *Lower bounds in Chebychev approximation from least-squares.* Consider the Chebychev approximation problem

$$\text{minimize } \|Ax-b\|_\infty \tag{33}$$

where $A \in \mathbf{R}^{m\times n}$ ($m \geq n$) and $\mathbf{Rank}\, A = n$. Let $x_{\rm cheb}$ denote an optimal point for the Chebychev approximation problem (there may be multiple optimal points; $x_{\rm cheb}$ denotes one of them).

The Chebychev problem has no closed-form solution, but the corresponding least-squares problem does. We denote the least-squares solution $x_{\rm ls}$ as

$$x_{\rm ls} = \operatorname{argmin} \|Ax-b\| = (A^TA)^{-1}A^Tb.$$

The question we address is the following. Suppose that for a particular A and b you have computed the least-squares solution $x_{\rm ls}$ (but not $x_{\rm cheb}$). How suboptimal is $x_{\rm ls}$ for the Chebychev problem? In other words, how much larger is $\|Ax_{\rm ls}-b\|_\infty$ than $\|Ax_{\rm cheb}-b\|_\infty$? To answer this question, we need a lower bound on $\|Ax_{\rm cheb}-b\|_\infty$.

(a) Prove the lower bound

$$\|Ax_{\rm cheb}-b\|_\infty \geq \frac{1}{\sqrt{m}}\|Ax_{\rm ls}-b\|_\infty,$$

using the fact that for all $y \in \mathbf{R}^m$,

$$\frac{1}{\sqrt{m}}\|y\| \leq \|y\|_\infty \leq \|y\|.$$

(b) In lecture 8 we derived the following dual for (33):

$$\begin{array}{ll} \text{maximize} & b^Tz \\ \text{subject to} & A^Tz = 0 \\ & \|z\|_1 \leq 1. \end{array} \tag{34}$$

We can use this dual problem to improve the lower bound obtained in (a).

- Denote the least-squares residual as $r_{\rm ls} = b - Ax_{\rm ls}$. Assuming $r_{\rm ls} \neq 0$, show that

  $$\hat{z} = -r_{\rm ls}/\|r_{\rm ls}\|_1, \quad \tilde{z} = r_{\rm ls}/\|r_{\rm ls}\|_1,$$

  are both feasible in (34).
- By duality $b^T\hat{z}$ and $b^T\tilde{z}$ are lower bounds for $\|Ax_{\rm cheb}-b\|_\infty$. Which is the better bound? How does it compare with the bound obtained in part (a) above?

One application is as follows. You need to solve the Chebychev approximation problem, but only within, say, 10%. You first solve the least-squares problem (which can be done faster), and then use the bound from part (b) to see if it can guarantee a maximum 10% error. If it can, great; otherwise solve the Chebychev problem (by slower methods).

Exercise 37. Let x be a real-valued random variable which takes values in $\{a_1, a_2, \ldots, a_n\}$ where $0 < a_1 < a_2 < \cdots < a_n$, and $\mathbf{Prob}(x = a_i) = p_i$. Obviously p satisfies $\sum_{i=1}^n p_i = 1$ and $p_i \geq 0$ for $i = 1, \ldots, n$.

(a) Consider the problem of determining the probability distribution that maximizes $\mathbf{Prob}(x \geq \alpha)$ subject to the constraint $\mathbf{E}\, x = b$, *i.e.*,

$$\begin{array}{ll} \text{maximize} & \mathbf{Prob}(x \geq \alpha) \\ \text{subject to} & \mathbf{E}\, x = b, \end{array} \qquad (35)$$

where α and b are given ($a_1 < \alpha < a_n$, and $a_1 \leq b \leq a_n$). The variable in problem (35) is the probability distribution, *i.e.*, the vector $p \in \mathbf{R}^n$. Write (35) as an LP.

(b) Take the dual of the LP in (a), and show that it is can be reformulated as

$$\begin{array}{ll} \text{minimize} & \lambda b + \nu \\ \text{subject to} & \lambda a_i + \nu \geq 0 \text{ for all } a_i < \alpha \\ & \lambda a_i + \nu \geq 1 \text{ for all } a_i \geq \alpha. \end{array}$$

The variables are λ and ν. Give a graphical interpretation of this problem, by interpreting λ and ν as coefficients of an affine function $f(x) = \lambda x + \nu$. Show that the optimal value is equal to

$$\begin{cases} \dfrac{b - a_1}{\bar{a} - a_1} & \text{if } b \leq \bar{a} \\ 1 & \text{if } b \geq \bar{a}, \end{cases}$$

where $\bar{a} = \min\{a_i \mid a_i \geq \alpha\}$. Also give the optimal values of λ and ν.

(c) From the dual solution, determine the distribution p that solves the problem in (a).

Exercise 38. This problem is a variation on the illumination problem of exercise 13. In part (a) of exercise 13 we formulated the problem

$$\begin{array}{ll} \text{minimize} & \max_{k=1,\ldots,n} |a_k^T p - I_{\text{des}}| \\ \text{subject to} & 0 \leq p \leq \mathbf{1} \end{array}$$

as the following LP in $p \in \mathbf{R}^m$ and an auxiliary variable w:

$$\begin{array}{ll} \text{minimize} & w \\ \text{subject to} & -w \leq a_k^T p - I_{\text{des}} \leq w, \quad k = 1, \ldots, n \\ & 0 \leq p \leq \mathbf{1}. \end{array} \qquad (36)$$

Now suppose we add the following constraint on the lamp powers p: no more than half the total power $\sum_{i=1}^m p_i$ is in any subset of r lamps (where r is a given integer with $0 < r < m$). The idea is to avoid solutions where all the power is concentrated in very few lamps. Mathematically, the constraint can be expressed as

$$\sum_{i=1}^{r} p_{[i]} \leq 0.5 \sum_{i=1}^{m} p_i \qquad (37)$$

where $p_{[i]}$ is the ith largest component of p. We would like to add this constraint to the LP (36). However the left-hand side of (37) is a complicated nonlinear function of p.

We can write the constraint (37) as a set of linear inequalities by enumerating all subsets $\{i_1, \ldots, i_r\} \subseteq \{1, \ldots, m\}$ with r different elements, and adding an inequality

$$\sum_{k=1}^{r} p_{i_k} \leq 0.5 \sum_{i=1}^{m} p_i$$

for each subset. Equivalently, we express (37) as

$$s^T p \leq 0.5 \sum_{i=1}^{m} p_i \text{ for all } s \in \{0,1\}^m, \ \sum_{i=1}^{m} s_i = r.$$

This yields a set of $\begin{pmatrix} m \\ r \end{pmatrix}$ linear inequalities in p.

We now use LP duality to derive a much more compact representation. We will prove that (37) can be expressed as the set of $1 + 2m$ linear inequalities

$$\begin{aligned} & rt + \sum_{i=1}^{m} x_i \leq 0.5 \sum_{i=1}^{m} p_i \\ & p_i \leq t + x_i, \quad i = 1, \ldots, m \\ & x \geq 0 \end{aligned} \tag{38}$$

in $p \in \mathbf{R}^m$, and auxiliary variables $x \in \mathbf{R}^m$ and $t \in \mathbf{R}$.

(a) Given a vector $p \in \mathbf{R}^m$, show that the sum of its r largest elements (*i.e.*, $p_{[1]} + \cdots + p_{[r]}$) is equal to the optimal value of the LP (in the variables $y \in \mathbf{R}^m$)

$$\begin{array}{ll} \text{maximize} & p^T y \\ \text{subject to} & 0 \leq y \leq \mathbf{1} \\ & \mathbf{1}^T y = r. \end{array} \tag{39}$$

(b) Derive the dual of the LP (39). Show that it can be written as

$$\begin{array}{ll} \text{minimize} & rt + \mathbf{1}^T x \\ \text{subject to} & t\mathbf{1} + x \geq p \\ & x \geq 0, \end{array} \tag{40}$$

where the variables are $t \in \mathbf{R}$ and $x \in \mathbf{R}^m$. By duality the LP (40) has the same optimal value as (39), *i.e.*, $p_{[1]} + \cdots + p_{[r]}$.

It is now clear that the optimal value of (40) is less than $0.5 \sum_i^m p_i$ if and only if there is a feasible solution t, x in (40) with $rt + \mathbf{1}^T x \leq 0.5 \sum_i^m p_i$. In other words, p satisfies the constraint (37) if and only if the set of linear inequalities (38) in x and t are feasible. To include the nonlinear constraint (37) in (36), we can add the inequalities (38), which yields

$$\begin{array}{ll} \text{minimize} & w \\ \text{subject to} & -w \leq a_k^T p - I_{\text{des}} \leq w, \quad k = 1, \ldots, n \\ & 0 \leq p \leq \mathbf{1} \\ & rt + \mathbf{1}^T x \leq 0.5 \ \ \mathbf{1}^T p \\ & p \leq t\mathbf{1} + x \\ & x \geq 0. \end{array}$$

This is an LP with $2m + 2$ variables p, x, w, t, and $2n + 4m + 1$ constraints.

Exercise 39. In this problem we derive a linear programming formulation for the following variation on ℓ^∞- and ℓ^1-approximation: given $A \in \mathbf{R}^{m\times n}$, $b \in \mathbf{R}^m$, and an integer k with $1 \le k \le m$,

$$\text{minimize } \sum_{i=1}^{k} |Ax - b|_{[i]}. \tag{41}$$

The notation $z_{[i]}$ denotes the ith largest component of $z \in \mathbf{R}^m$, and $|z|_{[i]}$ denotes the ith largest component of the vector $|z| = (|z_1|, |z_2|, \ldots, |z_m|) \in \mathbf{R}^m$. In other words in (41) we minimize the sum of the k largest residuals $|a_i^T x - b_i|$. For $k = 1$, this is the ℓ^∞-problem; for $k = m$, it is the ℓ^1-problem.

Problem (41) can be written as

$$\text{minimize } \max_{1\le i_1 < i_2 < \cdots < i_k \le m} \sum_{j=1}^{k} |a_{i_j}^T x - b_{i_j}|,$$

or as the following LP in x and t:

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & s^T(Ax - b) \le t \quad \text{for all } s \in \{-1, 0, 1\}^m,\ \|s\|_1 = k. \end{array}$$

Here we enumerate all vectors s with components -1, 0 or $+1$, and with exactly k nonzero elements. This yields an LP with $2^k \left(\begin{array}{c} m \\ k \end{array}\right)$ linear inequalities.

We now use LP duality to derive a more compact formulation.

(a) We have seen that for $c \in \mathbf{R}^m$ and $1 \le k \le n$, the the optimal value of the LP

$$\begin{array}{ll} \text{maximize} & c^T v \\ \text{subject to} & -y \le v \le y \\ & \mathbf{1}^T y = k \\ & y \le \mathbf{1} \end{array} \tag{42}$$

is equal to

$$|c|_{[1]} + \cdots + |c|_{[k]}.$$

Take the dual of the LP (42) and show that it can be simplified as

$$\begin{array}{ll} \text{minimize} & kt + \mathbf{1}^T z \\ \text{subject to} & -t\mathbf{1} - z \le c \le t\mathbf{1} + z \\ & z \ge 0. \end{array} \tag{43}$$

with variables $t \in \mathbf{R}$ and $z \in \mathbf{R}^m$. By duality the optimal values of (43) and (42) are equal.

(b) Now apply this result to $c = Ax - b$. From part (a), we know that the optimal value of the LP

$$\begin{array}{ll} \text{minimize} & kt + \mathbf{1}^T z \\ \text{subject to} & -t\mathbf{1} - z \le Ax - b \le t\mathbf{1} + z \\ & z \ge 0, \end{array} \tag{44}$$

with variables $t \in \mathbf{R}$, $z \in \mathbf{R}^m$ is equal to

$$\sum_{i=1}^{k} |Ax - b|_{[i]}.$$

Note that the constraints (44) are linear in x, so we can simultaneously optimize over x, *i.e.*, solve it as an LP with variables x, t and z. This way we can solve problem (41) by solving an LP with $m + n + 1$ variables and $3m$ inequalities.

Exercise 40. *A portfolio optimization problem.*

We consider a portfolio optimization problem with n assets or stocks held over one period. The variable x_i will denote the amount of asset i held at the beginning of (and throughout) the period, and p_i will denote the price change of asset i over the period, so the return is $r = p^T x$. The optimization variable is the portfolio vector $x \in \mathbf{R}^n$, which has to satisfy $x_i \geq 0$ and $\sum_{i=1}^n x_i \leq 1$ (unit total budget).

If p is exactly known, the optimal allocation is to invest the entire budget in the asset with the highest return, *i.e.*, if $p_j = \max_i p_i$, we choose $x_j = 1$, and $x_i = 0$ for $i \neq j$. However, this choice is obviously very sensitive to uncertainty in p. We can add various constraints to make the investment more robust against variations in p.

We can impose a diversity constraint that prevents us from allocating the entire budget in a very small number of assets. For example, we can require that no more than, say, 90% of the total budget is invested in any 5% of the assets. We can express this constraint as

$$\sum_{i=1}^{\lfloor n/20 \rfloor} x_{[i]} \leq 0.9$$

where $x_{[i]}$, $i = 1, \ldots, n$, are the values x_i sorted in decreasing order, and $\lfloor n/20 \rfloor$ is the largest integer smaller than or equal to $n/20$.

In addition, we can model the uncertainty in p by specifying a set $\mathcal{P}$ of possible values, and require that the investment maximizes the return in the worst-case scenario. The resulting problem is:

$$\begin{array}{ll} \text{maximize} & \min_{p \in \mathcal{P}} p^T x \\ \text{subject to} & \displaystyle\sum_{i=1}^{\lfloor n/20 \rfloor} x_{[i]} \leq 0.9 \\ & \mathbf{1}^T x \leq 1 \\ & x \geq 0. \end{array} \qquad (45)$$

For each of the following sets $\mathcal{P}$, can you express problem (45) as an LP?

(a) $\mathcal{P} = \{p^{(1)}, \ldots, p^{(K)}\}$, where $p^{(i)} \in \mathbf{R}^n$ are given. This means we consider a finite number of possible scenarios.

(b) $\mathcal{P} = \{\bar{p} + By \mid \|y\|_\infty \leq 1\}$ where $\bar{p} \in \mathbf{R}^n$ and $B \in \mathbf{R}^{n \times m}$ are given. We can interpret $\bar{p}$ as the expected value of p, and $y \in \mathbf{R}^m$ as uncertain parameters that determine the actual values of p.

(c) $\mathcal{P} = \{\bar{p} + y \mid By \leq d\}$ where $\bar{p} \in \mathbf{R}^n$, $B \in \mathbf{R}^{r \times m}$, and $d \in \mathbf{R}^r$ are given. Here we consider a polyhedron of possible value of p. (We assume that $\mathcal{P}$ is nonempty.)

You may introduce new variables and constraints, but you must clearly explain why your formulation is equivalent to (45). If you know more than one solution, you should choose the most compact formulation, *i.e.*, involving the smallest number of variables and constraints.

Exercise 41. *A generalized linear-fractional problem.*
Consider the problem

$$\begin{array}{ll} \text{minimize} & \dfrac{\|Ax-b\|_1}{c^Tx+d} \\ \text{subject to} & \|x\|_\infty \leq 1 \end{array} \tag{46}$$

where $A \in \mathbf{R}^{m\times n}$, $b \in \mathbf{R}^m$, $c \in \mathbf{R}^n$ and $d \in \mathbf{R}$ are given. We assume that $d > \|c\|_1$. As a consequence, $c^Tx+d > 0$ for all feasible x.

Remark: There are several correct answers to part (a), including a method based on solving one single LP.

(a) Explain how you would solve this problem using linear programming. If you know more than one method, you should give the simplest one.

(b) Prove that the following problem provides lower bounds on the optimal value of (46):

$$\begin{array}{ll} \text{maximize} & \lambda \\ \text{subject to} & \|A^Tz+\lambda c\|_1 \leq b^Tz - \lambda d \\ & \|z\|_\infty \leq 1. \end{array} \tag{47}$$

The variables are $z \in \mathbf{R}^m$ and $\lambda \in \mathbf{R}$.

(c) Use linear programming duality to show that the optimal values of (47) and (46) are in fact equal.

Exercise 42. Consider the problem

$$\text{minimize} \quad \sum_{i=1}^m h(a_i^Tx - b_i) \tag{48}$$

where h is the function

$$h(z) = \begin{cases} 0 & |z| \leq 1 \\ |z|-1 & |z| > 1 \end{cases}$$

and (as usual) $x \in \mathbf{R}^n$ is the variable, and $a_1, \ldots, a_m \in \mathbf{R}^n$ and $b \in \mathbf{R}^m$ are given. Note that this problem can be thought of as a sort of hybrid between ℓ^1 and ℓ^∞ approximation, since there is no cost for residuals smaller than one, and a linearly growing cost for residuals larger than one.

Express (48) as an LP, derive its dual, and simplify as much as you can.

Let x_{ls} denote the solution of the least-squares problem

$$\text{minimize} \quad \sum_{i=1}^m (a_i^Tx - b_i)^2,$$

and let r_{ls} denote the residual $r_{\text{ls}} = Ax_{\text{ls}} - b$. We assume A has rank n, so the least-squares solution is unique and given by

$$x_{\text{ls}} = (A^TA)^{-1}A^Tb.$$

The least-squares residual $r_{\rm ls}$ satisfies

$$A^T r_{\rm ls} = 0.$$

Show how to construct from $x_{\rm ls}$ and $r_{\rm ls}$ a feasible solution for the dual of (48), and hence a lower bound for its optimal value $p^\star$. Compare your lower bound with the trivial lower bound $p^\star \geq 0$. Is it always better, or only in certain cases?

Exercise 43. *Self-dual homogeneous LP formulation.*

(a) Consider the LP

$$\begin{array}{ll} \mbox{minimize} & f_1^T u + f_2^T v \\ \mbox{subject to} & M_{11} u + M_{12} v \leq f_1 \\ & -M_{12}^T u + M_{22} v = f_2 \\ & u \geq 0 \end{array} \tag{49}$$

in the variables $u \in \mathbf{R}^p$ and $v \in \mathbf{R}^q$. The problem data are the vectors $f_1 \in \mathbf{R}^p$, $f_2 \in \mathbf{R}^q$, and the matrices $M_{11} \in \mathbf{R}^{p\times p}$, $M_{12} \in \mathbf{R}^{p\times q}$, and $M_{22} \in \mathbf{R}^{q\times q}$.

Show that if M_{11} and M_{22} are *skew-symmetric*, *i.e.*,

$$M_{11}^T = -M_{11}, \quad M_{22}^T = -M_{22},$$

then the dual of the LP (49) can be expressed as

$$\begin{array}{ll} \mbox{maximize} & -f_1^T w - f_2^T y \\ \mbox{subject to} & M_{11} w + M_{12} y \leq f_1 \\ & -M_{12}^T w + M_{22} y = f_2 \\ & w \geq 0, \end{array} \tag{50}$$

with variables $w \in \mathbf{R}^p$ and $y \in \mathbf{R}^q$.

Note that the dual problem is essentially the same as the primal problem. Therefore if u, v are primal optimal, then $w = u$, $y = v$ are optimal in the dual problem. We say that the LP (49) with skew-symmetric M_{11} and M_{22} is *self-dual.*

(b) Write down the optimality conditions for problem (49). Use the observation we made in part (a) to show that the optimality conditions can be simplified as follows: u, v are optimal for (49) if and only if

$$\begin{aligned} M_{11} u + M_{12} v &\leq f_1 \\ -M_{12}^T u + M_{22} v &= f_2 \\ u &\geq 0 \\ u^T (f_1 - M_{11} u - M_{12} v) &= 0. \end{aligned}$$

In other words, u, v must be feasible in (49), and the nonnegative vectors u and

$$s = f_1 - M_{11} u - M_{12} v$$

must satisfy the complementarity condition $u^T s = 0$.

It can be shown that if (49) is feasible, then it has an optimal solution that is *strictly complementary*, *i.e.*,

$$u + s > 0.$$

(In other words, for each k either $s_k = 0$ or $u_k = 0$, but not both.)

(c) Consider the LP

$$
\begin{array}{ll}
\text{minimize} & 0 \\
\text{subject to} & b^T\tilde{z} + c^T\tilde{x} \leq 0 \\
& -b\tilde{t} + A\tilde{x} \leq 0 \\
& A^T\tilde{z} + c\tilde{t} = 0 \\
& \tilde{z} \geq 0, \quad \tilde{t} \geq 0
\end{array}
\tag{51}
$$

with variables $\tilde{x} \in \mathbf{R}^n$, $\tilde{z} \in \mathbf{R}^m$, and $\tilde{t} \in \mathbf{R}$. Show that this problem is self-dual. Use the result in part (b) to prove that (51) has an optimal solution that satisfies

$$\tilde{t}(c^T\tilde{x} + b^T\tilde{z}) = 0$$

and

$$\tilde{t} - (c^T\tilde{x} + b^T\tilde{z}) > 0.$$

Suppose we have computed an optimal solution with these properties. We can distinguish the following cases.

- $\tilde{t} > 0$. Show that $x = \tilde{x}/\tilde{t}$, $z = \tilde{z}/\tilde{t}$ are optimal for the pair of primal and dual LPs

$$
\begin{array}{ll}
\text{minimize} & c^T x \\
\text{subject to} & Ax \leq b
\end{array}
\tag{52}
$$

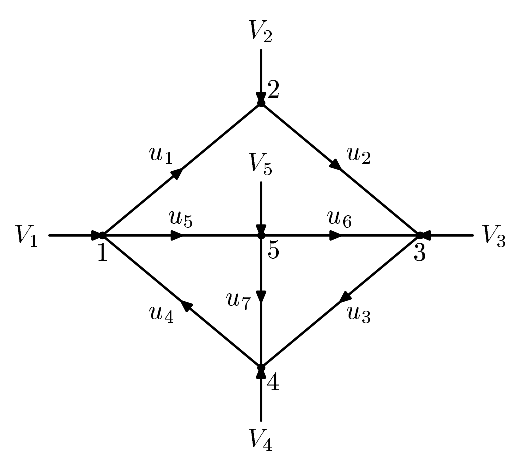
  and

$$
\begin{array}{ll}
\text{maximize} & -b^T z \\
\text{subject to} & A^T z + c = 0 \\
& z \geq 0.
\end{array}
\tag{53}
$$

- $\tilde{t} = 0$ and $c^T\tilde{x} < 0$. Show that the dual LP (53) is infeasible.
- $\tilde{t} = 0$ and $b^T\tilde{z} < 0$. Show that the primal LP (53) is infeasible.

This result has an important practical ramification. It implies that we do not have to use a two-phase approach to solve the LP (52) (*i.e.*, a phase-I to find a feasible point, followed by a phase-II to minimize $c^T x$ starting at the feasible point). We can solve the LP (52) and its dual, or detect primal or dual infeasibility, by solving one single, feasible LP (51). The LP (51) is much larger than (52), but it can be shown that the cost of solving it is not much higher if one takes advantage of the symmetry in the constraints.

Exercise 44. We consider a network flow problem on the simple network shown below:



Here $u_1, \ldots, u_7 \in \mathbf{R}$ denote the *flows* or traffic along links $1, \ldots, 7$ in the direction indicated by the arrow. (Thus, $u_1 = 1$ means a traffic flow of one unit in the direction of the arrow on link 1, *i.e.*, from node 1 to node 2.) $V_1, \ldots, V_5 \in \mathbf{R}$ denote the external inputs (or outputs if $V_i < 0$) to the network. We assume that the net flow into the network is zero, *i.e.*, $\sum_{i=1}^{5} V_i = 0$.

Conservation of traffic flow states that at each node, the total flow entering the node is zero. For example, for node 1, this means that $V_1 - u_1 + u_4 - u_5 = 0$. This gives one equation per node, so we have 5 traffic conservation equations, for the nodes $1, \ldots, 5$, respectively. (In fact, the equations are redundant since they sum to zero, so you could leave one, *e.g.*, for node 5, out. However, to answer the questions below, it is easier to keep all five equations.)

The cost of a flow pattern u is given by $\sum_i c_i |u_i|$, where $c_i > 0$ is the tariff on link i. In addition to the tariff, each link also has a maximum possible traffic level or link capacity: $|u_i| \leq U_i$.

(a) Express the problem of finding the minimum cost flow as an LP in inequality form, for the network shown above.

(b) Solve the LP from part (a) for the specific costs, capacities, and inputs

$$c = (2, 2, 2, 1, 1, 1, 1), \quad V = (1, 1, 0.5, 0.5, -3), \quad U = (0.5, 0.5, 0.1, 0.5, 1, 1, 1).$$

Find the optimal dual variables as well.

(c) Suppose we can increase the capacity of one link by a small fixed amount, say, 0.1. Which one should we choose, and why? (You're not allowed to solve new LPs to answer this!) For the link you pick, increase its capacity by 0.1, and then solve the resulting LP exactly. Compare the resulting cost with the cost predicted from the optimal dual variables of the original problem. Can you explain the answer?

(d) Now suppose we have the possibility to increase or reduce two of the five external inputs by a small amount, say, 0.1. To keep $\sum_i V_i = 0$, the changes in the two inputs must be equal in absolute value and opposite in sign. For example, we can increae V_1 by 0.1, and decrease V_4 by 0.1. Which two inputs should we modify, and why? (Again, you're not allowed to solve new LPs!) For the inputs you pick, change the value (increase or decrease, depending on which will result in a smaller cost) by 0.1, and then solve the resulting LP exactly. Compare the result with the one predicted from the optimal dual variables of the original problem.

Exercise 45. Let $P \in \mathbf{R}^{n \times n}$ be a matrix with the following two properties:

- all elements of P are nonnegative: $p_{ij} \geq 0$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n$
- the columns of P sum to one: $\sum_{i=1}^{n} p_{ij} = 1$ for $j = 1, \ldots, n$.

Show that there exists a $y \in \mathbf{R}^n$ such that

$$Py = y, \quad y \geq 0, \quad \sum_{i=1}^{n} y_i = 1.$$

Remark. This result has the following application. We can interpret P as the transition probability matrix of a Markov chain with n states: if $s(t)$ is the state at time t (*i.e.*, $s(t)$ is a random variable taking values in $\{1, \ldots, n\}$), then p_{ij} is defined as

$$p_{ij} = \mathbf{Prob}(s(t+1) = i \mid s(t) = j).$$

Let $y(t) \in \mathbf{R}^n$ be the probability distribution of the state at time t, *i.e.*,

$$y_i(t) = \mathbf{Prob}(s(t) = i).$$

Then the distribution at time $t+1$ is given by $y(t+1) = Py(t)$.

The result in this problem states that a finite state Markov chain always has an equilibrium distribution y.

Exercise 46. *Arbitrage and theorems of alternatives.*

Consider an event (for example, a sports game, political elections, the evolution of the stock-market over a certain period) with m possible outcomes. Suppose that n wagers on the outcome are possible. If we bet an amount x_j on wager j, and the outcome of the event is i ($i = 1, \ldots, m$), then our return will be equal to $r_{ij}x_j$ (this amount does not include the stake, *i.e.*, we pay x_j initially, and receive $(1+r_{ij})x_j$ if the outcome of the event is i, so $r_{ij}x_j$ is the net gain). We allow the bets x_j to be positive, negative, or zero. The interpretation of a negative bet is as follows. If $x_j < 0$, then initially we *receive* an amount of money $|x_j|$, with an obligation to *pay* $(1+r_{ij})|x_j|$ if outcome i occurs. In that case, we lose $r_{ij}|x_j|$, *i.e.*, our net is gain $r_{ij}x_j$ (a negative number).

We call the matrix $R \in \mathbf{R}^{m\times n}$ with elements r_{ij} the *return matrix*. A *betting strategy* is a vector $x \in \mathbf{R}^n$, with as components x_j the amounts we bet on each wager. If we use a betting strategy x, our total return in the event of outcome i is equal to $\sum_{j=1}^n r_{ij}x_j$, *i.e.*, the ith component of the vector Rx.

(a) *The arbitrage theorem.* Suppose you are given a return matrix R. Prove the following theorem: there is a betting strategy $x \in \mathbf{R}^n$ for which

$$Rx > 0 \tag{54}$$

if and only if there exists no vector $p \in \mathbf{R}^m$ that satisfies

$$R^T p = 0, \quad p \geq 0, \quad p \neq 0. \tag{55}$$

We can interpret this theorem as follows. If $Rx > 0$, then the betting strategy x guarantees a positive return for all possible outcomes, *i.e.*, it is a sure-win betting scheme. In economics, we say there is an *arbitrage opportunity.*

If we normalize the vector p in (55) so that $\mathbf{1}^T p = 1$, we can interpret it as a probability vector on the outcomes. The condition $R^T p = 0$ means that the expected return

$$\mathbf{E}\, Rx = p^T Rx = 0$$

for all betting strategies. We can therefore rephrase the arbitrage theorem as follows. There is no sure-win betting strategy (or arbitrage opportunity) if and only if there is a probability vector on the outcomes that makes all bets fair (*i.e.*, the expected gain is zero).

(b) *Options pricing.* The arbitrage theorem is a key result in mathematical finance, where it is used to determine prices of contracts. As a simple example, suppose we can invest in two assets: a stock and an option.

The current unit price of the stock is S. The price $\bar{S}$ of the stock at the end of the investment period is unknown, but it will be either $\bar{S} = Su$ or $\bar{S} = Sd$, where $u > 1$ and $d < 1$ are given numbers. In other words the price either goes up by a factor u, or down by a factor d. If the current interest rate over the investment period is r, then the present value of the stock price $\bar{S}$ at the end of the period is equal to $\bar{S}/(1+r)$, and our unit return is

$$\frac{Su}{1+r} - S = S\frac{u-r}{1+r}$$

if the stock goes up, and

$$\frac{Sd}{1+r} - S = S\frac{d-r}{1+r}$$

if the stock goes down.

We can also buy options, at a unit price of C. An option gives us the right to purchase one stock at a fixed price K at the end of the period. Whether we exercise the option or not, depends on the price of the stock at the end of the period. If the stock price $\bar{S}$ at the end of the period is greater than K, we exercise the option, buy the stock and sell it immediately, so we receive an amount $\bar{S}-K$. If the stock price $\bar{S}$ is less than K, we do not exercise the option and receive nothing. Combining both cases, we can say that the value of the option at the end of the period is $\max\{0, \bar{S}-K\}$, and the present value is $\max\{0, \bar{S}-K\}/(1+r)$. If we pay a price C per option, then our return is

$$\frac{1}{1+r}\max\{0, \bar{S}-K\} - C$$

per option.

We can summarize the situation with the return matrix

$$R = \begin{bmatrix} \dfrac{u-1-r}{1+r} & \dfrac{\max\{0, Su-K\}}{(1+r)C} - 1 \\[2ex] \dfrac{d-1-r}{1+r} & \dfrac{\max\{0, Sd-K\}}{(1+r)C} - 1 \end{bmatrix}.$$

The elements of the first row are the (present values of the) returns in the event that the stock price goes up. The second row are the returns in the event that the stock price goes down. The first column gives the returns per unit investment in the stock. The second column gives the returns per unit investment in the option.

In this simple example the arbitrage theorem allows us to determine the price of the option, given the other information S, K, u, d, and r. Show that if there is no arbitrage, then the price of the option C must be equal to

$$\frac{1}{1+r}(p\max\{0, Su-K\} + (1-p)\max\{0, Sd-K\})$$

where

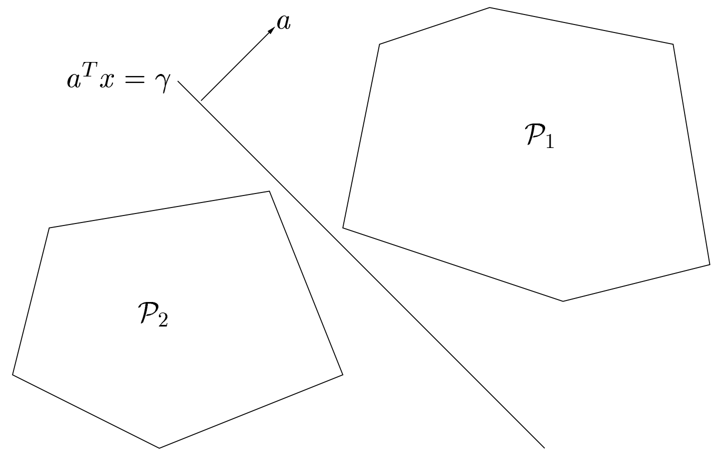
$$p = \frac{1+r-d}{u-d}.$$

Exercise 47. Describe a method for constructing a hyperplane that separates two given polyhedra

$$\mathcal{P}_1 = \{x \in \mathbf{R}^n | Ax \le b\}, \quad \mathcal{P}_2 = \{x \in \mathbf{R}^n | Cx \le d\}.$$

Your method must return a vector $a \in \mathbf{R}^n$ and a scalar γ such that

$$a^T x > \gamma \text{ for all } x \in \mathcal{P}_1, \quad a^T x < \gamma \text{ for all } x \in \mathcal{P}_2.$$



You can assume that $\mathcal{P}_1$ and $\mathcal{P}_2$ do not intersect. If you know several methods, you should give the most efficient one.

The simplex method

Exercise 48. Use the simplex method to solve the following LP:

$$\begin{array}{ll} \text{minimize} & -24x_1 + 396x_2 - 8x_3 - 28x_4 - 10x_5 \\ \text{subject to} & \begin{bmatrix} 12 & 4 & 1 & -19 & 7 \\ 6 & -7 & 18 & -1 & -13 \\ 1 & 17 & 3 & 18 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 12 \\ 6 \\ 1 \end{bmatrix} \\ & x \geq 0. \end{array}$$

Start with the initial basis $\{1, 2, 3\}$, and use Bland's rule to make pivot selections. Also compute the dual optimal point from the results of the algorithm.

Exercise 49. A square matrix is called *unimodular* if its elements are integer and its determinant is 0, 1 or -1. It follows from Cramer's rule that the inverse of a nonsingular unimodular matrix has integer components.

Let $\mathcal{P}$ be a polyhedron described by

$$Ax = b, \quad x \geq 0$$

where

- $A \in \mathbf{R}^{m \times n}$ has rank m and all $m \times m$ submatrices of A are unimodular,
- $b \in \mathbf{R}^m$ has integer components.

Show that all the extreme points of $\mathcal{P}$ have integer components.

Remark. This result allows us to solve certain classes of integer linear programs as ordinary LPs. As a simple example, consider the following problem. A manager in charge of N projects can choose among M contractors. Contractor j charges an amount of money c_{ij}

for project j, and can complete r_j projects within the available period of time, where r_j is a given integer. We would like to find the minimum cost assignment of projects to contractors.

We can formulate this problem as an LP with 0-1 constraints:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{N}\sum_{j=1}^{M} c_{ij}x_{ij} \\ \text{subject to} & \sum_{j=1}^{M} x_{ij} = 1, \quad i=1,\ldots,N \\ & \sum_{i=1}^{N} x_{ij} \leq r_j, \quad j=1,\ldots,M \\ & x_{ij} \in \{0,1\}, \quad i=1,\ldots,N, \;\; j=1,\ldots,M, \end{array} \tag{56}$$

where the variables x_{ij} have the following meaning:

$$\begin{array}{ll} x_{ij}=1 & \text{if project } i \text{ is assigned to contractor } j \\ x_{ij}=0 & \text{otherwise.} \end{array}$$

The first (equality) constraint in (56) states that each project should be assigned to exactly one contractor. The inequality constraints give the bounds on the number of projects each contractor can handle.

If we introduce slack variables y_j for the inequality constraints, we can write the problem as

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{N}\sum_{j=1}^{M} c_{ij}x_{ij} \\ \text{subject to} & \left[\begin{array}{ccccc} \mathbf{1}^T & 0 & \cdots & 0 & 0 \\ 0 & \mathbf{1}^T & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \mathbf{1}^T & 0 \\ I & I & \cdots & I & I \end{array}\right] \left[\begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_N \\ y \end{array}\right] = \left[\begin{array}{c} 1 \\ 1 \\ \vdots \\ 1 \\ r \end{array}\right] \\ & x_{ij} \in \{0,1\}, \quad y \geq 0, \end{array} \tag{57}$$

where $x_i = (x_{i1}, x_{i2}, \ldots, x_{iM})$ and $r = (r_1, \ldots, r_M)$. It can be shown that all submatrices in the coefficient matrix of (57) are unimodular, so the above result applies. Therefore we can replace the 0-1 constraints by inequalities $0 \leq x_{ij} \leq 1$ (or simply $0 \leq x_{ij}$, since the upper bound is implied by the equality constraints), and solve the assignment problem as an LP.

Interior-point methods

Exercise 50. In lecture 12 (page 12-17), we noted that a point $x^*(t)$ on the central path yields a dual feasible point

$$z_i^*(t) = \frac{1}{t(b_i - a_i^T x^*(t))}, \quad i=1,\ldots,m. \tag{58}$$

In this problem we examine what happens when $x^*(t)$ is calculated only approximately.

Suppose x is strictly feasible and v is the Newton step at x for the function

$$tc^T x + \phi(x) = tc^T x - \sum_{i=1}^{m} \log(b - a_i^T x).$$

Let $d \in \mathbf{R}^m$ be defined as $d_i = 1/(b_i - a_i^T x)$, $i = 1, \ldots, m$. Show that if

$$\lambda(x) = \|\,\mathbf{diag}(d)Av\| \leq 1,$$

then the vector

$$z = (d + \mathbf{diag}(d)^2 Av)/t$$

is dual feasible. Note that z coincides with (58) if $x = x^*(t)$ (and hence $v = 0$).

This observation is useful in a practical implementation of the barrier method. In practice, Newton's method provides an approximation of the central point $x^*(t)$, which means that the point (58) is not quite dual feasible, and a stopping criterion based on the corresponding dual bound is not quite accurate. The results derived above imply that even though $x^*(t)$ is not exactly centered, we can still obtain a dual feasible point, and use a completely rigorous stopping criterion.

Exercise 51. Let $\mathcal{P}$ be a polyhedron described by a set of linear inequalities:

$$\mathcal{P} = \{x \in \mathbf{R}^n \mid Ax \leq b\},$$

where $A \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$. As in lecture 12, ϕ denotes the logarithmic barrier function

$$\phi(x) = -\sum_{i=1}^{m} \log(b_i - a_i^T x).$$

(a) Suppose $\widehat{x}$ is strictly feasible. Show that

$$(x - \widehat{x})^T \nabla^2 \phi(\widehat{x})(x - \widehat{x}) \leq 1 \Longrightarrow Ax \leq b,$$

where $\nabla^2 \phi(\widehat{x})$ is the Hessian of ϕ at $\widehat{x}$. Geometrically, this means that the set

$$\mathcal{E}_{\text{inner}} = \left\{x \mid (x - \widehat{x})^T \nabla^2 \phi(\widehat{x})(x - \widehat{x}) \leq 1\right\},$$

which is an ellipsoid centered at $\widehat{x}$, is enclosed in the polyhedron $\mathcal{P}$.

(b) Suppose $\widehat{x}$ is the analytic center of the inequalities $Ax < b$. Show that

$$Ax \leq b \Longrightarrow (x - \widehat{x})^T \nabla^2 \phi(\widehat{x})(x - \widehat{x}) \leq m(m-1).$$

In other words, the ellipsoid

$$\mathcal{E}_{\text{outer}} = \left\{x \mid (x - \widehat{x})^T \nabla^2 \phi(\widehat{x})(x - \widehat{x}) \leq m(m-1)\right\}$$

contains the polyhedron $\mathcal{P}$.

Exercise 52. Let $\hat{x}$ be the analytic center of a set of linear inequalities

$$a_k^T x \leq b_k, \quad k = 1, \ldots, m.$$

Show that the kth inequality is redundant (*i.e.*, it can be deleted without changing the feasible set) if

$$b_k - a_k^T \hat{x} \geq m\sqrt{a_k^T H^{-1} a_k}$$

where H is defined as

$$H = \sum_{k=1}^{m} \frac{1}{(b_k - a_k^T \hat{x})^2} a_k a_k^T.$$

Exercise 53. The analytic center of a set of linear inequalities $Ax \leq b$ depends not only on the geometry of the feasible set, but also on the representation (*i.e.*, A and b). For example, adding redundant inequalities does not change the polyhedron, but it moves the analytic center. In fact, by adding redundant inequalities you can make *any* strictly feasible point the analytic center, as you will show in this problem.

Suppose that $A \in \mathbf{R}^{m\times n}$ and $b \in \mathbf{R}^m$ define a bounded polyhedron

$$\mathcal{P} = \{x \mid Ax \leq b\}$$

and that $x^\star$ satisfies $Ax^\star < b$. Show that there exist $c \in \mathbf{R}^n$, $\gamma \in \mathbf{R}$, and a positive integer q, such that

(a) $\mathcal{P}$ is the solution set of the $m + q$ inequalities

$$\begin{array}{l} Ax \leq b \\ \left.\begin{array}{l} c^T x \leq \gamma \\ c^T x \leq \gamma \\ \quad\vdots \\ c^T x \leq \gamma \end{array}\right\} q \text{ copies.} \end{array} \tag{59}$$

(b) $x^\star$ is the analytic center of the set of linear inequalities given in (59).

Exercise 54. *Potential reduction algorithm.* Consider the LP

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

with $A \in \mathbf{R}^{m\times n}$. We assume that $\mathbf{Rank}\, A = n$, that the problem is strictly feasible, and that the optimal value $p^\star$ is finite.

For $l < p^\star$ and $q > m$, we define the *potential function*

$$\varphi_{\text{pot}}(x) = q \log(c^T x - l) - \sum_{i=1}^m \log(b_i - a_i^T x).$$

The function φ_{pot} is defined for all strictly feasible x, and although it is not a convex function, it can be shown that it has a unique minimizer. We denote the minimizer as $x^\star_{\text{pot}}(l)$:

$$x^\star_{\text{pot}}(l) = \underset{Ax<b}{\operatorname{argmin}} \left(q \log(c^T x - l) - \sum_{i=1}^m \log(b_i - a_i^T x) \right).$$

(a) Show that $x^\star_{\text{pot}}(l)$ lies on the central path, *i.e.*, it is the minimizer of the function

$$tc^T x - \sum_{i=1}^m \log(b_i - a_i^T x)$$

for some value of t.

(b) Prove that the following algorithm converges and that it returns a suboptimal x with $c^T x - p^\star < \epsilon$.

> **given** $l < p^\star$, tolerance $\epsilon > 0$, $q > m$
>
> **repeat** {
>
> 1. $x := x^\star_{\rm pot}(l)$
> 2. **if** $\dfrac{m(c^Tx - l)}{q} < \epsilon$, **return**$(x)$
> 3. $l := \dfrac{q-m}{q}c^Tx + \dfrac{m}{q}l$
>
> }

Exercise 55. *The inverse barrier.*

The inverse barrier of a set of linear inequalities

$$a_i^Tx \le b_i, \quad i = 1, \dots, m,$$

is the function ψ, defined as

$$\psi(x) = \sum_{i=1}^m \frac{1}{b_i - a_i^Tx}$$

for strictly feasible x. It can be shown that ψ is convex and differentiable on the set of strictly feasible points, and that $\psi(x)$ tends to infinity as x approaches the boundary of the feasible set.

Suppose $\hat{x}$ is strictly feasible and minimizes

$$c^Tx + \psi(x).$$

Show that you construct from $\hat{x}$ a dual feasible point for the LP

$$\begin{array}{ll} \text{minimize} & c^Tx \\ \text{subject to} & a_i^Tx \le b_i, \; i = 1, \dots, m. \end{array}$$

Exercise 56. The most expensive step in one iteration of an interior-point method for an LP

$$\begin{array}{ll} \text{minimize} & c^Tx \\ \text{subject to} & Ax \le b \end{array}$$

is the solution of a set of linear equations of the form

$$A^TDA\,\Delta x = y, \tag{60}$$

where D is a positive diagonal matrix, the right-hand side y is a given vector, and Δx is the unknown. The values of D and y depend on the method used and on the current iterate, and are not important for our purposes here. For example, the Newton equation in the barrier method of lectures 12 and 13,

$$\nabla^2\phi(x)v = -tc - \nabla\phi(x),$$

is of the form (60), because $\nabla^2\phi(x)$ is given by the expression on page 12-11. In the primal-dual method of lecture 14, we have to solve two sets of linear equations of the form (60) with $D = X^{-1}Z$ (see page 14-15).

It is often possible to speed up the algorithm significantly by taking advantage of special structure of the matrix A when solving the equations (60).

Consider the following three optimization problems that we encountered before in this course.

- *ℓ_1-minimization*:

$$\text{minimize } \|Pu+q\|_1$$

($P \in \mathbf{R}^{r\times s}$ and $q \in \mathbf{R}^r$ are given; $u \in \mathbf{R}^s$ is the variable).

- *Constrained ℓ_1-minimization*:

$$\begin{array}{ll} \text{minimize} & \|Pu+q\|_1 \\ \text{subject to} & -\mathbf{1} \preceq u \preceq \mathbf{1} \end{array}$$

($P \in \mathbf{R}^{r\times s}$ and $q \in \mathbf{R}^r$ are given; $u \in \mathbf{R}^s$ is the variable).

- *Robust linear programming* (see exercise 18):

$$\begin{array}{ll} \text{minimize} & w^T u \\ \text{subject to} & Pu + \|u\|_1 \mathbf{1} \preceq q \end{array}$$

($P \in \mathbf{R}^{r\times s}$, $q \in \mathbf{R}^r$, and $w \in \mathbf{R}^s$ are given; $u \in \mathbf{R}^s$ is the variable).

For each of these three problems, answer the following questions.

(a) Express the problem as an LP in inequality form. Give the matrix A, and the number of variables and constraints.

(b) What is the cost of solving (60) for the matrix A you obtained in part (a), if you do not use any special structure in A (knowing that the cost of solving a general $n \times n$ linear system is $O(n^3)$ operations)?

(c) Work out the product $A^T DA$ (assuming D is a given positive diagonal matrix). Can you give an efficient method for solving (60) that uses the structure in the equations? What is the cost of your method (*i.e.*, the approximate number of operations when r and s are large) as a function of the dimensions r and s ?
Hint. Try to reduce the problem to solving a set of s linear equations in s variables, followed by a number of simple operations.
For the third problem, you can use the following formula for the inverse of a matrix $H + yy^T$, where y is a vector:

$$(H+yy^T)^{-1} = H^{-1} - \frac{1}{1+y^TH^{-1}y} H^{-1}yy^TH^{-1}.$$

Exercise 57. In this problem you are asked to write a Matlab code for the ℓ^1-approximation problem

$$\text{minimize } \|Pu+q\|_1, \tag{61}$$

where $P = \mathbf{R}^{r\times s}$ and $q \in \mathbf{R}^r$. The calling sequence for the code is `u = l1(P,q)`. On exit, it must guarantee a relative accuracy of 10^{-6} or an absolute accuracy of 10^{-8}, *i.e.*, the code can terminate if

$$\|Pu+q\|_1 - p^\star \leq 10^{-6} \cdot p^\star$$

or

$$\|Pu+q\|_1 - p^\star \leq 10^{-8},$$

where $p^\star$ is the optimal value of (61). You may assume that P has full rank ($\mathbf{Rank}\, P = s$).

We will solve the problem using Mehrotra's method as described in lecture 14, applied to the LP

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T v \\ \text{subject to} & \begin{bmatrix} P & -I \\ -P & -I \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \preceq \begin{bmatrix} -q \\ q \end{bmatrix}. \end{array} \tag{62}$$

We will take advantage of the structure in the problem to improve the efficiency.

(a) *Initialization.* Mehrotra's method can be started at infeasible primal and dual points. However good feasible starting points for the LP (62) are readily available from the solution $u_{\rm ls}$ of the least-squares problem

$$\text{minimize } \|Pu + q\|$$

(in Matlab: `u = -P\q`). As primal starting point we can use $u = u_{\rm ls}$, and choose v so that we have strict feasibility in (62). To find a strictly feasible point for the dual of (62), we note that $P^T P u_{\rm ls} = -P^T q$ and therefore the least-squares residual $r_{\rm ls} = Pu_{\rm ls} + q$ satisfies

$$P^T r_{\rm ls} = 0.$$

This property can be used to construct a strictly feasible point for the dual of (62). You should try to find a dual starting point that provides a positive lower bound on $p^\star$, *i.e.*, a lower bound that is better than the trivial lower bound $p^\star \geq 0$.

Since the starting points are strictly feasible, all iterates in the algorithm will remain strictly feasible, and we don't have to worry about testing the deviation from feasibility in the convergence criteria.

(b) As we have seen, the most expensive part of an iteration in Mehrotra's method is the solution of two sets of equations of the form

$$A^T X^{-1} Z A \Delta x = r_1 \tag{63}$$

where X and Z are positive diagonal matrices that change at each iteration. One of the two equations is needed to determine the affine-scaling direction; the other equation (with a different right-hand side) is used to compute the combined centering-corrector step. In our application, (63) has $r + s$ equations in $r + s$ variables, since

$$A = \begin{bmatrix} P & -I \\ -P & -I \end{bmatrix}, \quad \Delta x = \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}.$$

By exploiting the special structure of A, show that you can solve systems of the form (63) by solving a smaller system of the form

$$P^T D P \Delta u = r_2, \tag{64}$$

followed by a number of inexpensive operations. In (64) D is an appropriately chosen positive diagonal matrix. This observation is important, since it means that the cost of one iteration reduces to the cost of solving two systems of size $s \times s$ (as opposed to $(r+s) \times (r+s)$). In other words, although we have introduced r new variables to express (61) as an LP, the extra cost of introducing these variables is marginal.

(c) Test your code on randomly generated P and q. Plot the duality gap (on a logarithmic scale) versus the iteration number for a few examples and include a typical plot with your solutions.

Exercise 58. This problem is similar to the previous problem, but instead we consider the constrained ℓ^1-approximation problem

$$\begin{array}{ll} \text{minimize} & \|Pu + q\|_1 \\ \text{subject to} & -\mathbf{1} \leq u \leq \mathbf{1} \end{array} \tag{65}$$

where $P = \mathbf{R}^{r\times s}$ and $q \in \mathbf{R}^r$. The calling sequence for the code is `u = cl1(P,q)`. On exit, it must guarantee a relative accuracy of 10^{-5} or an absolute accuracy of 10^{-8}, *i.e.*, the code can terminate if

$$\|Pu+q\|_1 - p^\star \le 10^{-5}\cdot p^\star$$

or

$$\|Pu+q\|_1 - p^\star \le 10^{-8},$$

where $p^\star$ is the optimal value of (65). You may assume that P has full rank ($\mathbf{Rank}\,P = s$).

We will solve the problem using Mehrotra's method as described in lecture 14, applied to the LP

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T v \\ \text{subject to} & \begin{bmatrix} P & -I \\ -P & -I \\ I & 0 \\ -I & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \le \begin{bmatrix} -q \\ q \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}. \end{array} \tag{66}$$

We will take advantage of the structure in the problem to improve the efficiency.

(a) *Initialization.* For this problem it is easy to determine strictly feasible primal and dual points at which the algorithm can be started. This has the advantage that all iterates in the algorithm will remain strictly feasible, and we don't have to worry about testing the deviation from feasibility in the convergence criteria.

As primal starting point, we can simply take $u = 0$, and a vector v that satisfies $v_i > |(Pu+q)_i|$, $i = 1,\ldots,r$. What would you choose as dual starting point?

(b) As we have seen, the most expensive part of an iteration in Mehrotra's method is the solution of two sets of equations of the form

$$A^T X^{-1} Z A \Delta x = r_1 \tag{67}$$

where X and Z are positive diagonal matrices that change at each iteration. One of the two equations is needed to determine the affine-scaling direction; the other equation (with a different right-hand side) is used to compute the combined centering-corrector step. In our application, (67) has $r+s$ equations in $r+s$ variables, since

$$A = \begin{bmatrix} P & -I \\ -P & -I \\ I & 0 \\ -I & 0 \end{bmatrix}, \quad \Delta x = \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}.$$

By exploiting the special structure of A, show that you can solve systems of the form (67) by solving a smaller system of the form

$$(P^T \tilde{D} P + \hat{D})\Delta u = r_2, \tag{68}$$

followed by a number of inexpensive operations. The matrices $\tilde{D}$ and $\hat{D}$ in (68) are appropriately chosen positive diagonal matrices.

This observation is important, since the cost of solving (68) is roughly equal to the cost of solving the least-squares problem

$$\text{minimize } \|Pu+q\|.$$

Since the interior-point method converges in very few iterations (typically less than 10), this allows us to conclude that the cost of solving (65) is roughly equal to the cost of 10 least-squares problems of the same dimension, in spite of the fact that we introduced r new variables to cast the problem as an LP.

(c) Test your code on randomly generated P and q. Plot the duality gap (on a logarithmic scale) versus the iteration number for a few examples and include a typical plot with your solutions.