Convex Optimization
Homework 8

Spring 1401
Due date: 31st of Khordad

---

1. *Optimal spacecraft landing.* We consider the problem of optimizing the thrust profile for a spacecraft to carry out a landing at a target position. The spacecraft dynamics are

$$m\ddot{p} = f - mge_3,$$

where $m > 0$ is the spacecraft mass, $p(t) \in \mathbf{R}^3$ is the spacecraft position, with 0 the target landing position and $p_3(t)$ representing height, $f(t) \in \mathbf{R}^3$ is the thrust force, and $g > 0$ is the gravitational acceleration. (For simplicity we assume that the spacecraft mass is constant. This is not always a good assumption, since the mass decreases with fuel use. We will also ignore any atmospheric friction.) We must have $p\left(T^{\text{td}}\right) = 0$ and $\dot{p}\left(T^{\text{td}}\right) = 0$, where T^{td} is the touchdown time. The spacecraft must remain in a region given by

$$p_3(t) \geq \alpha \left\|(p_1(t), p_2(t))\right\|_2,$$

where $\alpha > 0$ is a given minimum glide slope. The initial position $p(0)$ and velocity $\dot{p}(0)$ are given. The thrust force $f(t)$ is obtained from a single rocket engine on the spacecraft, with a given maximum thrust; an attitude control system rotates the spacecraft to achieve any desired direction of thrust. The thrust force is therefore characterized by the constraint $\|f(t)\|_2 \leq F^{\max}$. The fuel use rate is proportional to the thrust force magnitude, so the total fuel use is

$$\int_0^{T^{td}} \gamma \|f(t)\|_2 dt$$

where $\gamma > 0$ is the fuel consumption coefficient. The thrust force is discretized in time, i.e., it is constant over consecutive time periods of length $h > 0$, with $f(t) = f_k$ for $t \in [(k-1)h, kh)$, for $k = 1, \ldots, K$, where $T^{\text{td}} = Kh$. Therefore we have

$$v_{k+1} = v_k + (h/m) f_k - hge_3, \quad p_{k+1} = p_k + (h/2)\left(v_k + v_{k+1}\right)$$

where p_k denotes $p((k-1)h)$, and v_k denotes $\dot{p}((k-1)h)$. We will work with this discrete-time model. For simplicity, we will impose the glide slope constraint only at the times $t = 0, h, 2h, \ldots, Kh$.

(a) Minimum fuel descent. Explain how to find the thrust profile $f_1, \ldots, f_K$ that minimizes fuel consumption, given the touchdown time $T^{\text{td}} = Kh$ and discretization time h.

(b) Minimum time descent. Explain how to find the thrust profile that minimizes the touchdown time, i.e., K, with h fixed and given. Your method can involve solving several convex optimization problems.

(c) Carry out the methods described in parts (a) and (b) above on the problem instance with data given in spacecraft_landing_data.*. Report the optimal total fuel consumption for part (a), and the minimum touchdown time for part (b). The data files also contain plotting code (commented out) to help you visualize your solution. Use the code to plot the spacecraft trajectory and thrust profiles you obtained for parts (a) and (b).

Hints.

- In Julia, the plot will come out rotated.

Remarks. If you'd like to see the ideas of this problem in action, watch these videos:

- http://www.youtube.com/watch?v=2t15vP1PyoA
- https://www.youtube.com/watch?v=orUjSkc2pGo

- https://www.youtube.com/watch?v=1B6oiLNyKKI
- https://www.youtube.com/watch?v=ZCBE8ocOkAQ

2. *Optimal racing of an energy-limited vehicle.* We have an energy-limited vehicle, such as a solar car, moving along a fixed straight track. We'd like to design a control system to move the vehicle from the starting point to the finishing point using minimum energy in the time interval $[0, T]$. (There are other related natural formulations of this problem, such as traversing the track in the minimum time subject to a maximum energy usage. We will not consider these here, but the same techniques are applicable.)

   At time t the car has position $x(t) \in \mathbf{R}$, velocity $v(t) \in \mathbf{R}$ and acceleration $a(t) \in \mathbf{R}$. The car starts with $x(0) = 0$ and $v(0) = 0$ and must finish with $x(T) \geq x^{\text{final}}$.

   At time t the kinetic energy of the vehicle is $k(t) = \frac{1}{2}mv(t)^2$, where m is the mass. Let the energy delivered from the battery to the drivetrain be $p(t)$, which is nonnegative (there is no regenerative braking.) Then

   $$\dot{k}(t) = p(t) - p^{\text{brake}}(t) - p^{\text{loss}}(t)$$

   where $p^{\text{brake}}(t) \geq 0$ is an input that the control system (i.e., your optimization) chooses, and losses due to drag are modeled via

   $$p^{\text{loss}}(t) = c^{\text{loss}} v(t)^3$$

   Here c^{loss} is a positive constant that depends on the shape of the vehicle and the density of the air, The vehicle must move according to the following requirements. Tire traction limits acceleration so that $\dot{v}(t) \leq a^{\max}$. Note that there is no lower bound on the acceleration. The vehicle cannot move backwards and must stay within the speed limit, and so $0 \leq v(t) \leq v^{\max}$. The final velocity of the vehicle must satisfy $v(T) \leq v^{\text{final}}$.

   We will use period $h > 0$ and sample position according to $x_i = x(ih)$, and similarly for velocity, acceleration and kinetic energy. The vehicle dynamics $\dot{x}(t) = v(t)$ and $\dot{v}(t) = a(t)$ are then discretized according to

   $$x_{i+1} = x_i + \frac{h}{2}(v_i + v_{i+1}), \qquad v_{i+1} = v_i + ha_i$$

   and the rate of change of kinetic energy is discretized according to

   $$\frac{1}{h}(k_{i+1} - k_i) = p_i - p_i^{\text{brake}} - p_i^{\text{loss}}$$

   We would like to minimize the total energy used, which is discretized as

   $$E = h\sum_{i=0}^{n} p_i$$

   where $T = nh$. The parameters are

   $$m = 10 \quad x^{\text{final}} = 10 \quad v^{\text{final}} = 1 \quad v^{\max} = 10 \quad a^{\max} = 2 \quad c^{\text{loss}} = 2 \quad h = 0.1 \quad T = 5$$

   (a) Formulate this problem as an optimization problem with variables p_i, x_i, v_i, k_i (and others if necessary) for $i = 0, \ldots, n$. If this problem is not convex, explain briefly why.

   (b) By relaxing the energy constraint $k(t) = \frac{1}{2}mv(t)^2$ to

   $$k(t) \geq \frac{1}{2}mv(t)^2$$

   state a convex optimization problem whose solution provides an optimal trajectory x, v, p, and k for part (a). Explain why the relaxation is tight. By tight, we mean that the solution to your problem has the same optimal value as that of part (a).

   (c) Carry out your method from part (b). Report the optimal value of the total energy E. Plot the position x, velocity v and power used p of the vehicle as functions of time.

3. *Simple portfolio optimization.* We consider a portfolio optimization problem as described on pages 155 and 185-186 of Convex Optimization, with data that can be found in the file simple_portfolio_data.*.

   (a) Find minimum-risk portfolios with the same expected return as the uniform portfolio ($x = (1/n)\mathbf{1}$), with risk measured by portfolio return variance, and the following portfolio constraints (in addition to $\mathbf{1}^T x = 1$):

   - No (additional) constraints.
   - Long-only: $x \succeq 0$.
   - Limit on total short position: $\mathbf{1}^T (x_-) \leq 0.5$, where $(x_-)_i = \max\{-x_i, 0\}$.

   Compare the optimal risk in these portfolios with each other and the uniform portfolio.

   (b) Plot the optimal risk-return trade-off curves for the long-only portfolio, and for total shortposition limited to 0.5, in the same figure. Follow the style of figure 4.12 (top), with horizontal axis showing standard deviation of portfolio return, and vertical axis showing mean return.

4. *Maximizing algebraic connectivity of a graph.* Let $G = (V, E)$ be a weighted undirected graph with $n = |V|$ nodes, $m = |E|$ edges, and weights $w_1, \ldots, w_m \in \mathbf{R}_+$on the edges. If edge k connects nodes i and j, then define $a_k \in \mathbf{R}^n$ as $(a_k)_i = 1, (a_k)_j = -1$, with other entries zero. The weighted Laplacian (matrix) of the graph is defined as

$$L = \sum_{k=1}^{m} w_k a_k a_k^T = A \operatorname{diag}(w) A^T,$$

where $A = [a_1 \cdots a_m] \in \mathbf{R}^{n \times m}$ is the incidence matrix of the graph. Nonnegativity of the weights implies $L \succeq 0$. Denote the eigenvalues of the Laplacian L as

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n,$$

which are functions of w. The minimum eigenvalue λ_1 is always zero, while the second smallest eigenvalue λ_2 is called the algebraic connectivity of G and is a measure of the connectedness of a graph: The larger λ_2 is, the better connected the graph is. It is often used, for example, in analyzing the robustness of computer networks.

Though not relevant for the rest of the problem, we mention a few other examples of how the algebraic connectivity can be used. These results, which relate graph-theoretic properties of G to properties of the spectrum of L, belong to a field called spectral graph theory. For example, $\lambda_2 > 0$ if and only if the graph is connected. The eigenvector v_2 associated with λ_2 is often called the Fiedler vector and is widely used in a graph partitioning technique called spectral partitioning, which assigns nodes to one of two groups based on the sign of the relevant component in v_2. Finally, λ_2 is also closely related to a quantity called the isoperimetric number or Cheeger constant of G, which measures the degree to which a graph has a bottleneck.

The problem is to choose the edge weights $w \in \mathbf{R}_+^m$, subject to some linear inequalities (and the nonnegativity constraint) so as to maximize the algebraic connectivity:

$$\begin{array}{ll} \text{maximize} & \lambda_2 \\ \text{subject to} & w \succeq 0, \quad Fw \preceq g, \end{array}$$

with variable $w \in \mathbf{R}^m$. The problem data are A (which gives the graph topology), and F and g (which describe the constraints on the weights).

   (a) Describe how to solve this problem using convex optimization.

   (b) Numerical example. Solve the problem instance given in max_alg_conn_data.m, which uses $F = \mathbf{1}^T$ and $g = 1$ (so the problem is to allocate a total weight of 1 to the edges of the graph). Compare the algebraic connectivity for the graph obtained with the optimal weights $w^\star$ to the one obtained with $w^{\text{unif}} = (1/m)\mathbf{1}$ (i.e., a uniform allocation of weight to the edges).

   Use the function plotgraph (A, xy, w) to visualize the weighted graphs, with weight vectors $w^\star$ and w^{unif}. You will find that the optimal weight vector $v^\star$ has some zero entries (which due to the finite precision of the solver, will appear as small weight values); you may want to round small values (say, those under 10^{-4}) of $w^\star$ to exactly zero. Use the gplot function to visualize the original (given) graph, and the subgraph associated with nonzero weights in $w^\star$. Briefly comment on the following (incorrect) intuition: "The more edges a graph has, the more connected it is, so the optimal weight assignment should make use of all available edges."

5. *Radiation treatment planning.* In radiation treatment, radiation is delivered to a patient, with the goal of killing or damaging the cells in a tumor, while carrying out minimal damage to other tissue. The radiation is delivered in beams, each of which has a known pattern; the level of each beam can be adjusted. (In most cases multiple beams are delivered at the same time, in one 'shot', with the treatment organized as a sequence of 'shots'.) We let b_j denote the level of beam j, for $j = 1, \ldots, n$. These must satisfy $0 \leq b_j \leq B^{\max}$, where $B^{\max}$ is the maximum possible beam level. The exposure area is divided into m voxels, labeled $i = 1, \ldots, m$. The dose d_i delivered to voxel i is linear in the beam levels, i.e., $d_i = \sum_{j=1}^{n} A_{ij} b_j$. Here $A \in \mathbf{R}_+^{m \times n}$ is a (known) matrix that characterizes the beam patterns. We now describe a simple radiation treatment planning problem.

   A (known) subset of the voxels, $\mathcal{T} \subset \{1, \ldots, m\}$, corresponds to the tumor or target region. We require that a minimum radiation dose D^{target} be administered to each tumor voxel, i.e., $d_i \geq D^{\text{target}}$ for $i \in \mathcal{T}$. For all other voxels, we would like to have $d_i \leq D^{\text{other}}$, where D^{other} is a desired maximum dose for non-target voxels. This is generally not feasible, so instead we settle for minimizing the penalty
$$E = \sum_{i \notin \mathcal{T}} \left(\left(d_i - D^{\text{other}} \right)_+ \right)^2,$$
   where $(\cdot)_+$denotes the nonnegative part. We can interpret E as the sum of the squares of the nontarget excess doses.

   (a) Show that the treatment planning problem is convex. The optimization variable is $b \in \mathbf{R}^n$; the problem data are $B^{\max}, A, \mathcal{T}, D^{\text{target}}$, and D^{other} .

   (b) Solve the problem instance with data given in the file treatment_planning_data.m. Here we have split the matrix A into Atarget, which contains the rows corresponding to the target voxels, and Aother, which contains the rows corresponding to other voxels. Give the optimal value. Plot the dose histogram for the target voxels, and also for the other voxels. Make a brief comment on what you see. Remark. The beam pattern matrix in this problem instance is randomly generated, but similar results would be obtained with realistic data.

6. **Optional.** *Ranking by aggregating preferences.* We have n objects, labeled $1, \ldots, n$. Our goal is to assign a real valued rank r_i to the objects. A preference is an ordered pair (i, j), meaning that object i is preferred over object j. The ranking $r \in \mathbf{R}^n$ and preference (i, j) are consistent if $r_i \geq r_j + 1$. (This sets the scale of the ranking: a gap of one in ranking is the threshold for preferring one item over another.) We define the preference violation of preference (i, j) with ranking $r \in \mathbf{R}^n$ as
$$v = (r_j + 1 - r_i)_+ = \max\{r_j + 1 - r_i, 0\}.$$
   We have a set of m preferences among the objects, $\left(i^{(1)}, j^{(1)}\right), \ldots, \left(i^{(m)}, j^{(m)}\right)$. (These may come from several different evaluators of the objects, but this won't matter here.)

   We will select our ranking r as a minimizer of the total preference violation penalty, defined as
$$J = \sum_{k=1}^{m} \phi\left(v^{(k)}\right)$$
   where $v^{(k)}$ is the preference violation of $\left(i^{(k)}, j^{(k)}\right)$ with r, and ϕ is a nondecreasing convex penalty function that satisfies $\phi(u) = 0$ for $u \leq 0$.

   (a) Make a (simple, please) suggestion for ϕ for each of the following two situations:

      i. We don't mind some small violations, but we really want to avoid large violations.

ii. We want as many preferences as possible to be consistent with the ranking, but will accept some (hopefully, few) larger preference violations.

(b) Find the rankings obtained using the penalty functions proposed in part (a), on the data set found in rank_aggr_data.m. Plot a histogram of preference violations for each case and briefly comment on the differences between them. Give the number of positive preference violations for each case. (Use sum $(v > 0.001)$ to determine this number.)

Remark. The objects could be candidates for a position, papers at a conference, movies, websites, courses at a university, and so on. The preferences could arise in several ways. Each of a set of evaluators provides some preferences, for example by rank ordering a subset of the objects. The problem can be thought of as aggregating the preferences given by the evaluators, to come up with a composite ranking.

7. **Optional.** *Optimizing processor speed.* A set of n tasks is to be completed by n processors. The variables to be chosen are the processor speeds $s_1, \ldots, s_n$, which must lie between a given minimum value $s_{\min}$ and a maximum value $s_{\max}$. The computational load of task i is α_i, so the time required to complete task i is $\tau_i = \alpha_i / s_i$.

The power consumed by processor i is given by $p_i = f(s_i)$, where $f : \mathbf{R} \to \mathbf{R}$ is positive, increasing, and convex. Therefore, the total energy consumed is

$$E = \sum_{i=1}^{n} \frac{\alpha_i}{s_i} f(s_i).$$

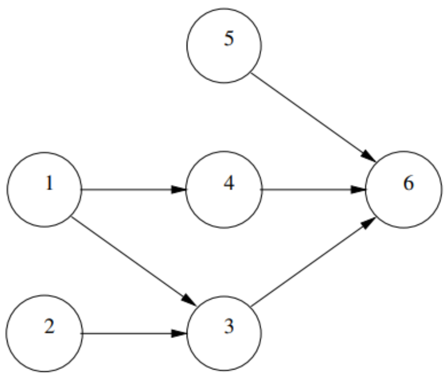
(Here we ignore the energy used to transfer data between processors, and assume the processors are powered down when they are not active.)

There is a set of precedence constraints for the tasks, which is a set of m ordered pairs $\mathcal{P} \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\}$. If $(i, j) \in \mathcal{P}$, then task j cannot start until task i finishes. (This would be the case, for example, if task j requires data that is computed in task i.) When $(i, j) \in \mathcal{P}$, we refer to task i as a precedent of task j, since it must precede task j. We assume that the precedence constraints define a directed acyclic graph (DAG), with an edge from i to j if $(i, j) \in \mathcal{P}$.

If a task has no precedents, then it starts at time $t = 0$. Otherwise, each task starts as soon as all of its precedents have finished. We let T denote the time for all tasks to be completed.

To be sure the precedence constraints are clear, we consider the very small example shown below, with $n = 6$ tasks and $m = 6$ precedence constraints.

$$\mathcal{P} = \{(1,4), (1,3), (2,3), (3,6), (4,6), (5,6)\}$$



In this example, tasks 1,2 , and 5 start at time $t = 0$ (since they have no precedents). Task 1 finishes at $t = \tau_1$, task 2 finishes at $t = \tau_2$, and task 5 finishes at $t = \tau_5$. Task 3 has tasks 1 and 2 as precedents, so it starts at time $t = \max\{\tau_1, \tau_2\}$, and ends τ_3 seconds later, at $t = \max\{\tau_1, \tau_2\} + \tau_3$. Task 4 completes at time $t = \tau_1 + \tau_4$. Task 6 starts when tasks 3,4 , and 5 have finished, at time $t = \max\{\max\{\tau_1, \tau_2\} + \tau_3, \tau_1 + \tau_4, \tau_5\}$. It finishes τ_6 seconds later. In this example, task 6 is the last task to be completed, so we have

$$T = \max\{\max\{\tau_1, \tau_2\} + \tau_3, \tau_1 + \tau_4, \tau_5\} + \tau_6$$

(a) Formulate the problem of choosing processor speeds (between the given limits) to minimize completion time T, subject to an energy limit $E \leq E_{\max}$, as a convex optimization problem. The data in this problem are $\mathcal{P}, s_{\min}, s_{\max}, \alpha_1, \ldots, \alpha_n, E_{\max}$, and the function f. The variables are $s_1, \ldots, s_n$.

Feel free to change variables or to introduce new variables. Be sure to explain clearly why your formulation of the problem is convex, and why it is equivalent to the problem statement above.

Important:

- Your formulation must be convex for any function f that is positive, increasing, and convex. You cannot make any further assumptions about f.
- This problem refers to the general case, not the small example described above.

(b) Consider the specific instance with data given in proc_speed_data.m, and processor power

$$f(s) = 1 + s + s^2 + s^3.$$

The precedence constraints are given by an $m \times 2$ matrix prec, where m is the number of precedence constraints, with each row giving one precedence constraint (the first column gives the precedents).

Plot the optimal trade-off curve of energy E versus time T, over a range of T that extends from its minimum to its maximum possible value. (These occur when all processors operate at $s_{\max}$ and $s_{\min}$, respectively, since T is monotone nonincreasing in s.) On the same plot, show the energy-time trade-off obtained when all processors operate at the same speed $\bar{s}$, which is varied from $s_{\min}$ to $s_{\max}$.

Note: In this part of the problem there is no limit $E^{\max}$ on E as in part (a); you are to find the optimal trade-off of E versus T.

Good Luck!