MATHEMATICAL TOOLS - PROBLEM SET 11

Due Sunday, January 29th, 23:55, either in the course mailbox or through the Moodle.

Problem 1. Let $A \in M_{m,n}(\mathbb{R}), b \in \mathbb{R}^m$. Show that if there exists $c \in \mathbb{R}^m$ s.t. $c^T A \geq 0$ (i.e. every coordinate is non-negative) and $c^T b < 0$, then $\{x \in \mathbb{R}^n : x \geq 0, Ax = b\} = \emptyset$.

Problem 2. Let $a_1, a_2, \ldots, a_9$ be the nine digits of your ID number. Consider the following linear program: Maximize:

$$x_1 + 2x_2 + 3x_3$$

Subject to:

$$\begin{aligned} a_1x_1 + a_2x_2 + a_3x_3 &\leq 1 \\ a_4x_1 + a_5x_2 + a_6x_6 &\leq 1 \\ a_7x_1 + a_8x_2 + a_9x_3 &\leq 1 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

Find an optimal solution (and prove that it is optimal), or else prove that an optimal solution doesn't exist. You may use any method you desire. For example, you might use the simplex method described in recitation, but you can also use any other algorithm (or computer program), so long as you submit a proof that your answer is correct.

Problem 3. Let $A \in M_{m,n}(\mathbb{R}), m \leq n$ have rank m and let $b \in \mathbb{R}^m$. Recall that $y \in \mathbb{R}^n$ is called a *basic feasible solution* to the linear program $Ax = b, x \geq 0$ if there exists $B \in \binom{[n]}{m}$ s.t.

- A_B is non-singular (where A_B is the $m \times m$ matrix obtained by taking the columns of A corresponding to indices in B).
- $supp(y) \subseteq B$ (i.e. $y_i \neq 0 \implies i \in B$).
- $Ay = b, y \geq 0$ (i.e. y is a *feasible solution*).

(1) Let $B \in \binom{[n]}{m}$ and assume A_B is non-singular. Show that there exists a unique $x \in \mathbb{R}^n$ s.t. $Ax = b$ and $supp(x) \subseteq B$. Is x necessarily a basic feasible solution?

(2) A point z in a convex set C is called *extreme* if there are no two points $x \neq y \in C$ s.t. $z = \lambda x + (1-\lambda)y, \lambda \in (0,1)$. Prove that x is an extreme point of the polyhedron $P = \{y \in \mathbb{R}^n : y \geq 0, Ay = b\}$ iff x is a basic feasible solution.

(3) Prove that if the linear program $\max c^T x, x \in P$ has an optimal solution then it has an optimal solution which is a basic feasible solution.

(4) **Integral Solutions**: You saw in class that integer programming is NP-hard. However, in some special cases, we can solve integer programming problems efficiently. Here is a criterion for when this is possible.
Let $A \in M_{m,n}(\mathbb{Z}), m \leq n$ have rank m. Let $b \in \mathbb{Z}^m$. Assume that for every square submatrix C of A, $\det C \in \{-1, 0, 1\}$ (matrices with this property

are called **totally unimodular**). Let $c \in \mathbb{R}^n$. Show that if there is an optimal solution to the linear program $\max c^T x, Ax = b, x \geq 0$, then there is an optimal solution with integral coordinates.

Problem 4. Max-Flow as Linear Programming: A *flow-network* is a (finite) directed graph $G = (V, E)$with a capacity function $c : E \to [0, \infty)$ and two special vertices $s, t \in V$. For $v \in V$, define $I(v) = \{e \in E : \exists u \in V, e = (u, v)\}$ and $O(v) = \{e \in E : \exists u \in V, e = (v, u)\}$ (these are the ingoing and outgoing edges, respectively). A *flow* in G is a function $f : E \to \mathbb{R}$ satisfying:

- Capacity constraints: $\forall e \in E, 0 \leq f(e) \leq c(e)$
- Flow conservation constraints: $\forall v \in V \backslash \{s, t\}, \sum_{e \in I(v)} f(e) = \sum_{e \in O(v)} f(e)$.

You've probably seen the min-cut max-flow theorem, which states that the maximal flow in G is equal to its minimum cut. Additionally, the proof of the MCMC theorem usually also proves that if the capacities are integral, then the maximal flow may be takin to be integral. Linear programming is a natural framework in which to prove both of these facts. Unfortunately, the first part is usually done with duality, which you won't cover in class until next week. Here we'll prove the part of the theorem concerning integral flows.

(1) Write a linear program that maximizes the total flow out of s (i.e. maximizes $\phi(f) = \sum_{e \in O(s)} f(e) - \sum_{e \in I(s)} f(e)$). You should make use of G's *vertex-edge* incidence matrix: This is a $|V \setminus \{s, t\}| \times |E|$ matrix, with a 1 in every entry (v, e) where e is an outgoing edge of v, and a -1 in every (v, e) where e is an ingoing edge of v (all other entries are 0).

(2) Show that there is a maximal flow (i.e. that there exist feasible solutions and the program is bounded).

(3) Show that if all the capacities are integral then there is a maximal flow with integral values.
Guidance: Write a linear program in the form $Ax = b, x \geq 0$, with b integral, and show that A is totally unimodular. Then apply part 4 of the previous question. In order to prove total unimodularity, you'll need to think about the structure of A. It's only non-zero entries should be ± 1. Also, one of the following should always hold: Either there is a row or column with a single non-zero entry, or there is an all-zero row or column, or every column has exactly one 1 and exactly one -1.