Problem Set 10

Due: December 2, 1999.

Problem 1. We proved that using a separation algorithm for a polytope P you could optimize over P. Here we prove that if you can optimize then you can separate. Assume that $0 \in P$. Define the *polar operator* $*$ for polytopes P by

$$P^* = \{z \mid zx \leq 1 \ \forall x \in P\}$$

(a) Show that $P^{**} = P$

(b) Show that given an optimization algorithm for P, we can separate over P^* (hint: to separate w from P^*, optimize $\max\{wx \mid x \in P\}$

(b) deduce that given an optimization algorithm for P, we can separate over P (hint: take a trip through the polar—remember that we know how to optimize if we can separate)

Problem 2. In this problem, you will fill in the details of an interior-point type algorithm for a restricted class of linear programs known as **packing and covering** problems.

Consider the linear program

$$\min \sum_j x_j$$

$$\begin{aligned} \sum_j a_{ij} x_j &\geq 1 \quad \forall i \\ x_j &\geq 0 \quad \forall j \end{aligned}$$

and its dual

$$\max \sum_i y_i$$

$$\begin{aligned} \sum_i a_{ij} y_i &\leq 1 \\ x_i &\geq 0. \end{aligned}$$

Assume that $A = [a_{ij}]$ is $m \times n$ and has only *nonnegative* entries. The primal problem is a **covering** problem: you want to use the smallest amount of x to cover each constraint by one unit. The dual is a **packing** problem: you want to pack in as much y as possible, subject to not overfilling any constraint.

You will now show that a continuous algorithm solves (almost miraculously) the above pair of dual linear programs. We shall define a series of functions whose argument is the "time" and you'll show that some of these functions tend to the optimal solution as time goes to infinity. (For simplicity of notation, we drop the dependence on the time.)

- Initially, we let $s_j = 0$ for $j = 1, \ldots, n$ and $LB = 0$. The vector s will (sort of) play the role of primal solution, and LB the role of a lower bound on the objective function.
- At any time, let
$$t_i = e^{-\sum_j a_{ij} s_j}$$
for $i = 1, \ldots, m$. Also, let $d_j = \sum_i a_{ij} t_i$ for $j = 1, \ldots, n$, $D = \max_j d_j$ and k be an index j attaining the maximum in the definition of D. The algorithm continuously increases s_k.

Observe that when s_k is increased, the vectors t and d as well as D change also, implying that the index k changes over time.

(a) Let $\alpha = \min_i(\sum_j a_{ij} s_j)$. Let $x_j = s_j/\alpha$ for $j = 1, \ldots, n$, $y_i = t_i/D$ for $i = 1, \ldots, m$ and $LB = \max(LB, \frac{\sum t_i}{D})$. Show that x is primal feasible, y is dual feasible and LB is a lower bound on the optimal value of both primal and dual.

(b) Prove that
$$\sum_{i=1}^{m} t_i \leq m e^{-\sum_{j=1}^{n} s_j / LB}.$$
Hint: Show that initially the inequality holds and that it is also maintained whenever we have equality.

(c) Deduce from (b) that $\sum_i t_i$ tends to 0 as time goes to infinity.

(d) Using (b), give an upper bound on the value of the primal solution x, and using (c), show that this upper bound tends to LB as time goes to infinity. This shows that as time goes to infinity, both x and y tend to primal and dual optimal solutions!

Problem 3. The lion-hunting techniques I distributed by email show a marked focus on historical (mathematical/physical) techniques. Develop a new approach that exploits our modern understanding of algorithmic efficiency.