CS 573: Graduate Algorithms, Fall 2008
Homework 6

Practice only

- This homework is only for practice; do not submit solutions. At least one (sub)problem (or something *very* similar) will appear on the final exam.

---

1. An *integer program* is a linear program with the additional constraint that the variables must take only integer values.

   (a) Prove that deciding whether an integer program has a feasible solution is NP-complete.

   (b) Prove that finding the optimal solution to an integer program is NP-hard.

   [Hint: Almost any NP-hard decision problem can be formulated as an integer program. Pick your favorite.]

2. Describe precisely how to dualize a linear program written in general form:

$$\begin{aligned}
\text{maximize } & \sum_{j=1}^{d} c_j x_j \\
\text{subject to } & \sum_{j=1}^{d} a_{ij} x_j \le b_i \quad \text{for each } i = 1..p \\
& \sum_{j=1}^{d} a_{ij} x_j = b_i \quad \text{for each } i = p+1..p+q \\
& \sum_{j=1}^{d} a_{ij} x_j \ge b_i \quad \text{for each } i = p+q+1..n
\end{aligned}$$

   Keep the number of dual variables as small as possible. The dual of the dual of any linear program should be *syntactically identical* to the original linear program.

3. Suppose you have a subroutine that can solve linear programs in polynomial time, but only if they are both feasible and bounded. Describe an algorithm that solves *arbitrary* linear programs in polynomial time, using this subroutine as a black box. Your algorithm should return an optimal solution if one exists; if no optimum exists, your algorithm should report that the input instance is UNBOUNDED or INFEASIBLE, whichever is appropriate. *[Hint: Add one constraint to guarantee boundedness; add one variable to guarantee feasibility.]*

4. Suppose you are given a set P of n points in some high-dimensional space $\mathbb{R}^d$, each labeled either *black* or *white*. A *linear classifier* is a d-dimensional vector c with the following properties:

   - If p is a black point, then $p \cdot c > 0$.
   - If p is a while point, then $p \cdot c < 0$.

   Describe an efficient algorithm to find a linear classifier for the given data points, or correctly report that none exists. *[Hint: This is almost trivial, but not quite.]*

Lots more linear programming problems can be found at http://www.ee.ucla.edu/ee236a/homework/problems.pdf. Enjoy!