Massachusetts Institute of Technology
6.854J/18.415J: Advanced Algorithms
David Karger

Handout 13
Wednesday, October 18, 2005

Problem Set 7

Due: Wednesday, October 26, 2005.

Problem 1. Suppose you are given two polyhedra $P = \{x \mid Ax \le b\}$ and $Q = \{x \mid Dx \le e\}$.

(a) Using duality, prove that if polyhedra P and Q have empty intersection (i.e., no point is in both), then there are $y, z \ge 0$ such that $yA + zD = 0$ but $yb + ze < 0$.

(b) Conclude that if polyhedra P and Q have empty intersection (i.e., no point is in both), then there is a separating hyperplane for P and Q (i.e., a vector c such that $c \cdot x < c \cdot w$ for all $x \in P$ and $w \in Q$). **Hint:** consider $c = yA$ from the previous part.

(c) Conclude that given the two polyhedra, there is a quickly verifiable answer as to whether or not the two polyhedra have a point in common (do not worry about whether the numbers in your answer have large representations).

NONCOLLABORATIVE Problem 2. Another way to formulate the maximum-flow problem as a linear program is via flow decomposition. Suppose we consider all (exponentially many) s-t paths P in the network G, and let f_P be the amount of flow on path P. Then maximum flow says to find

$$\begin{aligned} z &= \max \sum f_P, \\ \sum_{P \ni e} f_P &\le u_e, \\ f_P &\ge 0. \end{aligned}$$

(The first constraint says that the total flow on all paths through e must be less than u_e.) Take the dual of this linear program and give an English explanation of the objective and constraints.

Problem 3. Although the dual can tell you a lot about the structure of a problem, knowing an optimal dual solution does not in general help you solve the primal problem. Suppose we had an algorithm that could optimize an LP with an $m \times n$ constraint matrix in $O((m+n)^k)$ time given an optimal solution to the dual LP.

(a) Argue that any LP optimization problem can be transformed into the following form:

$$\begin{array}{ll} \text{minimize} & 0 \cdot x \\ \text{subject to} & Ax = b \\ & x \geq 0 \end{array}$$

(This LP has optimum value 0 if it is feasible, and ∞ if it is infeasible.)

(b) What is the dual of this linear program?

(c) Argue that, if the primal is feasible, then the dual has an obvious optimum solution.

(d) Deduce that, given the algorithm above, you can build an LP algorithm that will solve any LP *without* knowing a dual solution, in the same asymptotic time bounds as the algorithm above.

Problem 4. Consider a graph in which edges have costs (possibly negative, representing profits). Suppose you want to find a *minimum mean cycle* in this graph: one with the minimum ratio of cost to length (number of edges). Going around such a cycle repeatedly (assuming it is negative) provides you with the maximum possible profit per unit length/time, so is the fastest way to earn money if you are, for example, a delivery service. Minimum mean cycle also arises as a subroutine for solving problems like min cost flow. Consider the following linear program:

$$\begin{aligned} w &= \min \sum c_{ij} f_{ij} \\ \sum_j f_{ij} - f_{ji} &= 0 \quad (\forall i) \\ \sum f_{ij} &= 1 \\ f_{ij} &\geq 0 \end{aligned}$$

(a) Explain why this captures the minimum mean cycle problem. (**Hint:** f_{ij} is a circulation so can be decomposed into cycles.)

(b) Give the dual of this linear program—it will involve maximizing a certain variable λ.

(c) Give an explanation (in terms of min-cost-flow reduced costs) for why this dual formulation also captures minimum mean cycles. (**Hint:** how much is added to the cost of a k-edge cycle?)

(d) Let's assume the costs c_{ij} are integers. Suggest a combinatorial algorithm (not based on linear programming) that uses binary search to find the right λ to solve the dual problem. Can you use this to find a minimum mean cycle? **Note:** to know when you can terminate the search, you will need to lower bound the difference between the smallest and next smallest mean cost of a cycle.