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

Handout 17
November 16, 1999

Problem Set 9

Due: November 23, 1999.

Problem 1. 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 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 2. As was discussed briefly in class, the strongly polynomial min-cost flow algorithms work by finding a *minimum mean cost cycle*—that is, a cycle minimizing the ratio of cost to number of edges. Consider the following linear program:

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

(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 reduced costs) for why this formulation also captures minimum mean cycles (hint: how much is added to the cost of a k-edge cycle?)

(d) 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.

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 LP algorithm that could optimize an LP with an $m \times n$ constraint matrix in $O((m+n)^k)$ time.

- Argue that any LP optimization problem can be transformed into the following form: $\min\{0 \mid Ax = b,\ x \geq 0\}$. (This LP has optimum value 0 if it is feasible, and ∞ if it is infeasible.)
- What is the dual of this linear program?
- Argue that if the primal is feasible, the dual has an obvious optimum solution.
- Deduce that given the above algorithm, you can build an LP algorithm that will solve any LP **without** knowing a dual solution in the same asymptotic time bounds.

Problem 4. Markov chains. An $n \times n$ matrix P is **stochastic** if all entries are nonnegative and every row sums to 1, that is $\sum_j p_{ij} = 1$ (so each row can be thought of as taking a convex combination). Stochastic matrices are used to represent the **transition matrices** of **Markov chains**—random walks through a series of states. The term p_{ij} represents the probability, if you are in a current state i, that your next state will be j (thus the sum to one rule). If you have a probability distribution π over your current state, where π_i dentoes the probability you are in state i, then after a transition with probability defined by P, your new probability distribution is πP.

Use duality (or Farkas' Lemma) to prove that for any stochastic matrix P, there is a **nonzero** $\pi \geq 0$ such that $\pi P = \pi$. The vector π can be normalized to 1, in which case it represents a probability distribution that is **stationary** under the action of the transition matrix—that is, if π is the probability distribution on what state you are in before a transition, it is also the probability distribution after the transition. This proves that every Markov chain has a stationary probability distribution.

Hint: you must somehow express the constraint $\pi > 0$ (a strict inequality). Consider the constraint $\sum \pi_i = 1$.

Problem 5. Submit a $\frac{1}{2}$- to 1-page writeup of your project plan, on a page separate from your problem set. The writeup can be the same as your other group members', but each person should submit one. It should include references.

****Problem 6.** Game theory (this problem is neat, and not really hard, but you are working too hard so it is optional). In a *0-sum 2-player game*, Alice has a choice of n so-called *pure* strategies and Bob has a choice of m *pure* strategies. If Alice picks strategy i and Bob picks strategy j, then the *payoff* is a_{ij}, meaning a_{ij} dollars are transfered from Alice to Bob. So Bob makes money if a_{ij} is positive, but Alice makes money if a_{ij} is negative. Thus, Alice wants to pick a strategy that minimizes the payoff while Bob wants a strategy that maximizes the payoff. The matrix $A = (a_{ij})$ is called the *payoff matrix*.

It is well known that to play these games well, you need to use a *mixed* strategy—a random choice from among pure strategies. A mixed strategy is just a particular probability distribution over pure strategies: you flip coins and then play the selected pure strategy. If Alice has mixed strategy x, meaning he plays strategy i with probability x_i, and Bob has mixed strategy y, then it is easy to prove that the expected payoff in the resulting game is xAy. Alice wants to minimize this expected payoff while Bob wants to maximize it. Our goal is to understand what strategies each player should play.

We'll start by making the pessimal assumption for Alice that whichever strategy she picks, Bob will play best possible strategy against her. In other words, given Alice's strategy x, Bob will pick a strategy y that achieves $\max_y xAy$. Thus, Alice wants to find a distribution x that minimizes $\max_y xAy$. Similarly, Bob wants a y to maximize $\min_x xAy$.

So we are interested in solving the following 2 problems:

$$\min_{\sum x_i = 1} \max_{\sum y_j = 1} xAy$$
$$\max_{\sum y_j = 1} \min_{\sum x_i = 1} xAy$$

Unfortunately, these are nonlinear programs!

(a) Show how to convert each program above into a linear program, and thus find an optimal strategy for both players in polynomial time.

(b) Give a plausible explanation for the meaning of your linear program (why does it give the optimum?)

(c) Use strong duality (applied to the LP you built in the previous part) to argue that the above two quantities are *equal*.

The second statement shows that the strategies x and y, besides being optimal, are in *Nash Equilibrium*: even if each player knows the other's strategy, there is no point in changing strategies. This was proven by Von Neumann and was actually one of the ideas that led to the discovery of strong duality.