CMSC 858S: Randomized Algorithms

Fall 2001

Ungraded Homework Assignment #2

This is the second *ungraded* homework assignment. The suggested time period to finish this assignment is 1 week. We see from Chebyshev's inequality that the variance of a random variable lets us get reasonable tail bounds for it. The first two problems are about the variance, and about tail bounds via second moment methods. The third problem is about the notion of *flow decomposition*. It will enable us to focus on the "paths-specified" version of the edge-disjoint paths problem that we discussed in class, instead of the original (no paths specified) version. The fourth problem is on an application of Chernoff-type bounds to complexity theory.

1. (a) The *covariance* of random variables X and Y, denoted $\mathrm{covar}[X, Y]$, is defined to be $\mathbf{E}[XY] - \mathbf{E}[X] \cdot \mathbf{E}[Y]$. (Note that $\mathrm{covar}[X, X]$ is the variance $\mathrm{var}[X]$ of a random variable X.) Show that

$$\mathrm{var}[X_1 + X_2 + \ldots + X_n] = (\sum_{i=1}^{n} \mathrm{var}[X_i]) + 2 \sum_{i<j} \mathrm{covar}[X_i, X_j].$$

(b) Show that if random variable X takes on values only in $[0, 1]$, then $\mathrm{var}[X] \leq \mathbf{E}[X]$. (**Hint:** Use convexity similarly as in our derivation of the Chernoff-Hoeffding bounds.)

(c) Use parts (a) and (b) to give an alternative proof of the following fact shown in class: if $X = \sum_{i=1}^{n} X_i$ where the X_i are independent random variables each taking values in $[0, 1]$, then $\mathrm{var}[X] \leq \mathbf{E}[X]$.

2. Problem 3.5(a) in the Motwani-Raghavan book. (**Hint:** Apply Markov's inequality to a suitable quadratic polynomial in X.)

3. Suppose G is a directed graph, let s and t be two vertices in it, and let $a \geq 0$ be a real. (All the notions here can be extended to undirected graphs G, by replacing each edge of G by two anti-parallel arcs.) As we know, a *flow f of value a from s to t in G* is an assignment of non-negative values ("flows") to the arcs of G such that: (i) the net outgoing flow from s and the net incoming flow for t equal a; (ii) the total incoming flow equals the total outgoing flow for all other vertices. An *(ℓ, a)–path flow f from s to t in G* is a collection of *at most ℓ* paths P_i such that each P_i is a path from s to t in G, along with an assignment of a non-negative value b_i to each P_i such that $\sum_i b_i = a$.

(i) Suppose we are given a flow f of value $a > 0$ from s to t in G; let m denote the number of arcs in G. Give an efficient algorithm to find an (m, a)–path flow f' from s to t in G such that using the above notation for path flows, we have for each arc e that the sum of b_i over all paths P_i that contain e, equals the flow assigned to e by the given flow f. This process of converting a flow to a path flow is called *flow decomposition*. (**Hint:** Choose a path from s to t which has all arcs with positive–i.e., nonzero–flow values. Try to use the smallest flow value on the arcs of this path in some way.)

(ii) Write an LP relaxation for the "no paths specified" version of the edge-disjoint paths problem, using the notion of flows. Assume that the input graph is a directed graph $G = (V, E)$. Observe that flow decomposition applied to these flows, gives a feasible solution to the LP relaxation of a "paths specified" version of the given ("no paths specified") instance; the former is what we studied in class.

4. Problem 4.8 in the Motwani-Raghavan book.