Advanced Algorithms: Assignment 3

November 12, 2019

Instructions:

(i) Discussions or consulting references are not encouraged. If at all you discuss with anyone or consult a reference, you should clearly mention it.

(ii) Copying others' assignments is **strictly** prohibited.

1. In class, we have seen a $\frac{1}{2}(\sqrt{5}-1)$-approximation algorithm for MAX SAT using biased coins and a randomized $\left(1-\frac{1}{e}\right)$-approximation using randomized LP rounding. Show that choosing the better of the two solutions gives a 0.75-approximation for MAX SAT.

2. In the analysis of network reliability estimation where the failure probability p_{fail} is $\Omega(\frac{1}{poly(n)})$, we performed t independent experiments involving removing each edge of the network with probability p and checking if the resulting graph is disconnected. We defined a random variable X_i to be 1 if the ith experiment resulted in a disconnected graph and 0 otherwise. (Recall that p is the failure probability for each edge of the network.) Then the fraction of the disconnected subgraphs (i.e. $X = \frac{1}{t}\sum_{i=1}^{t} X_i$)served as a reasonable estimate to the failure probability. The number of experiments t required to get a $(1 \pm \epsilon)$-estimate of p_{fail} with probability $1-\delta$ is $\frac{\sigma^2}{\delta\epsilon^2\mu^2}$, where $\mu = E[X_i]$ and $\sigma^2 = Var[X_i]$. Thus, for fixed ϵ, μ, σ^2, we require $t = \Omega(\frac{1}{\delta})$. As a result, if δ is required to be exponentially small, we need exponentially many experiments.

   So we execute the above algorithm s times, replacing δ with $\frac{1}{4}$ for each run (find this s as part of your analysis) and take the median of all the outcomes. Show that the number of experiments $t' = ts = \Omega(\log\frac{1}{\delta})$.

3. We have seen an FPRAS for counting the number of satisfying assignments to a DNF. Generalize this to *probabilistic DNF* where each variable is set to 1 with probability p, independently and the goal is to estimate the probability that the given DNF formula is satisfied. *Note that this is an essential ingredient in the network reliability estimation problem we have seen in class.*

4. **Randomized load balancing:** In class, we have seen approximation algorithms for the multiprocessor scheduling problem. Consider the scenario when the number of processors n is very large, the running time of jobs is not known in advance, a randomized algorithm simply assigns a randomly chosen processor to each of the m jobs independent of the other jobs. This can be seen as throwing m balls into n bins where $m >> n$. Answer the following:

   Note: You may need Stirling's approximation.

   (a) What is the probability of two balls falling into the same bin (called *collision*)?
   (b) What is the expected number of collisions?
   (c) What is the probability that a particular bin is empty? What is the expected number of empty bins?
   (d) What is the probability that a particular bin has at least k balls?

(e) Using the above, show that with high probability i.e. with probability $1 - \frac{1}{n}$, all bins have at most $\frac{3 \ln n}{\ln \ln n}$ balls.

(f) Show that the expected maximum number of balls in any bin is $\frac{\ln n}{\ln \ln n}(1 + O(1))$.

A significant improvement by the power of two choices: In the above, instead of choosing one processor independently and uniformly at random (i.u.a.r.) for each job, if we choose two processors i.u.a.r. for each job, and assign the job to the less loaded processor between the two, then the expected maximum number of jobs is $\frac{\ln \ln n}{\ln 2} + O(1)$.