Advanced Algorithms – Exercise 2

Deadline: 15/12/2005

1. In class we saw a 2-approximation algorithm for the MAX-CUT problem. In this question we will design a *randomized* approximation algorithm for this problem, with the same approximation guarantee.

   We will first devise an algorithm A that will provide a 2-approximation in expectation. We will then derive from A another algorithm B that will obtain a result arbitrarily close to the expectation of A, with arbitrarily high probability.

   The first algorithm A goes as follows: each vertex is equally likely to be in one of two sets: V_1 or V_2. A returns the cut (V_1, V_2).

   (a) Denote by ALG_A the random variable that indicates the size of the cut produced by A, for a given instance. Show that $E[ALG_A] \geq \frac{OPT}{2}$.

   (b) Show that for any constant $\epsilon > 0$,
   $$\Pr[ALG_A \geq (\frac{1}{2} - \epsilon)OPT] \geq \frac{\epsilon}{2}$$

   (c) For any constants $\delta, \epsilon > 0$, devise an approximation algorithm B such that
   $$\Pr[ALG_B \geq (\frac{1}{2} - \epsilon)OPT] < 1 - \delta$$
   where ALG_B is the random variable that indicates the size of the cut produced by B. Analyze the running time of B in terms of the input size, δ, and ϵ.

2. Derandomize the 2-approximation algorithm for MAX-CUT designed in the previous question using the method of conditional expectations. Show that the derandomized algorithm is similar to the deterministic algorithm taught in class.

   Remark: It is enough to consider the first algorithm A.

3. In class we saw an approximation algorithm for the problem of finding integer multi-commodity flow. We have proved it for the special case where the minimum capacity was $O(\log m)$. In this question we will suppose that all edges have capacities of 1, and show a $O(\sqrt{m})$-approximation algorithm for this case.

   Recall that for each commodity i we found paths $p_1^i, ..., p_n^i$, and associated each one with a weight $t_1^1, ..., t_n^1$. We then chose at most one path such that the path p_j^i was chosen with probability $\frac{t_j^i}{1+\epsilon}$, for some $\epsilon > 0$. The change in the new algorithm is that the path p_j^i will be chosen with probability $\frac{t_j^i}{O(\sqrt{m})}$.

   Prove that this is indeed an $O(\sqrt{m})$-approximation algorithm for the problem (you need to specify the constants inside the O.) Use the following version of the chernoff bounds:

Claim: Let $X_1, ... X_l$ be independent random variables that take values in $[0,1]$ and $R = \Sigma_{s\in[l]} X_s$. If $\mu = E[X]$ and $\delta \geq 0$ it holds that $Pr[R \geq \mu(1+\delta)] \leq (\frac{e}{(1+\delta)})^{-(1+\delta)\mu}$

4. Suppose we throw n identical balls to n different bins, where each ball is thrown to a bin with equal probability.

   (a) Show that with constant probability no bin contains more than $t = O(\log n)$ balls. Calculate the value of the constants inside the O.

   Remark: A more careful analysis yields that $t = O(\frac{\log n}{\log \log n})$.

   (b) Use the Chebishev inequality to derive another (worse) bound for the value of t.

   Remark: The difference between the values of the two t's calculated in this question is because the chernoff bounds show that the probability of deviating from the mean decreases exponentially, while Chebishev only guarantees that the same probability fall off polynomially.

   (c) Now suppose that we throw $n \log n$ identical balls to n different bins. Show that with constant probability no bin contains more than $O(\log n)$ balls. Calculate the value of the constants inside the O.