Introduction to approximation and randomized algorithms

3rd exercise session 13th November 2019

Exercise 1. Let us have a biased coin, where we get heads with an unknown probability p. But we would like to have a fair coin. Are you able to simulate a flip of a fair coin by using a few flips of the biased coin? What is the expected number of the biased coin flips to simulate one flip of a fair coin?

Exercise 2. Now consider the oposite problem. We are given a fair coin and we want to simulate a flip of a biased coin, where we get heads with probability p. In each of the following cases find a way how to simulate the needed flip and also compute the expected number of fair coin flips to simulate one biased coin flip.

a) First assume, that $p = \ell/2^k$ for $k,\ell \in \mathbb{N}$ such that $0 < \ell < 2^k$.
b) Try to extend the previous algorithm to work for any rational p, that is a p of a form $p = a/b$ where $a,b \in \mathbb{N}$
c) Finally, try to find an algorithm that works for all values of p, that is even for irrational p.

Exercise 3. We'll look at the problem of maximum k-cut now. In this problem, we are given an undirected graph $G = (V,E)$ with nonnegative edge wights w_e. Our goal is to partition the vertex set V into groups $V_1, \ldots, V_k$ in such a way, that the sum of edge weights of edges going between different groups is maximized. Try to find a randomized $\frac{k-1}{k}$-approximation algorithm for this problem. If you would find it easier, you might look first at the variant for $k = 2$, which is the standard maximum cut.

Exercise 4. Now consider the problem of minimum dominating set. On input we are given an undirected graph $G = (V,E)$ and we want to find a set $U \subseteq V$ of minimum size such that each vertex $v \in V$ is dominated by some vertex $u \in U$. We say that a vertex v is dominated by a vertex u, if $v = u$ or v is a neighbor of u. This problem is NP-hard, so we would like to find a good approximation. In this exercise our goal is $O(\log n)$-approximation. We're also going to restrict ourselves to d-regular graphs. Follow these steps:

1. First show that if we'd be able to find a dominating set of size $\frac{cn \log n}{d+1}$ in polynomial time, then we would get an $(c \log n)$-approximation algorithm.

2. Algorithm will choose $k = \frac{cn \log n}{d+1}$ vertices from V uniformly at random (for suitable c which we're going to choose later) and inserts them into the set U. To simplify the analysis, we allow ourselves to choose the same vertex multiple times. Our goal now is to show that a set chosen in this way is a dominating set with a high probability. First find a probability that vertex $v \in V$ is dominated by the i-th chosen vertex.

3. The next step is to find a probability, that vertex $v \in V$ is not dominated by any chosen vertex.

4. Finally find the probability that U is a dominating set. How do we choose the value of c?