Problem Set 1 (due Friday, April 20)

Problem 1. (Percolation on 3-D grid)

Give any non-trivial bounds on the critical probability for percolation on the 3-D grid. That is, present explicit values for $p_1 > 0$ and $p_2 < 1$ such that the critical probability for percolation on the 3-D grid is in $[p_1, p_2]$. Prove your claim.

Problem 2. (Analysis of greedy online learning)

In the online learning problem we studied in class, we have n experts, each giving a binary prediction for an event in each step. Suppose we consider a greedy procedure for combining their predictions: in each step, we select the prediction of the expert that has the highest correct predictions until that step.

(a) Suppose we break ties using a deterministic strategy. Show that an adversary can ensure that the ratio of the number of mistakes made by the online learning algorithm to the number of mistakes made by the best expert can be made to be at least n.

(b) Suppose we break ties randomly; that is, we pick an expert uniformly at random from the set of experts that have the highest correct predictions until that step. Show that the number of mistakes made by the randomized greedy algorithm is at most $O(\lg n)$ times the number of mistakes made by the best expert. (It may be instructive to try showing that this bound is, in fact, asymptotically tight for the algorithm).

Note that the weighted majority algorithm achieved a $(1 + \varepsilon)$-approximation for $\varepsilon > 0$ that can be made arbitrarily small, for a long enough sequence of steps.

Problem 3. (Maximum cut with additional constraint)

Consider the Max-cut problem with the additional constraint that for two given sets S_1 and S_2 of pairs of vertices, we need to ensure that every pair of vertices in S_1 is on the same side of the cut, while every pair of vertices in S_2 is separated by the cut. Write new quadratic program and vector programs for this constrained maximum cut problem and show how to modify Goemans-Williamson's algorithm to achieve the same approximation factor as for Max-Cut.

Problem 4. (Unsplittable multicommodity flow)

In the multicommodity flow problem we defined in class, we assumed that a flow from a source to a sink can be split among multiple paths. In some applications, especially in the networking domain, one may demand flow of a single commodity to be sent along a single path. We call this the unsplittable multicommodity flow problem.

(a) Show that the unsplittable multicommodity flow problem is NP-hard, even with two commodities.

(*Hint:* Reduce from the partition problem.)

Consider the special case of the demands version of the unsplittable multicommodity flow problem in which the demand of each commodity as well as the capacity of each edge is 1. We study the following randomized rounding algorithm for unsplittable multicommodity flow in this case.

1. Solve the LP relaxation of the problem (which removes the unsplittability constraint).
2. Decompose the flow of each commodity i into a set P_i of at most m flow paths, where m is the number of edges. For each commodity i, select a path at random from P_i with probability equal to the flow of i along the path. Send all the flow for commodity i (as determined by the LP) along this path. Call the resulting flow f.
3. Return the flow obtained by scaling f down by the largest amount of flow going across any edge.

(b) Prove that after step 2 of the above algorithm, the expected flow along any edge is at most 1.

(c) Prove that after step 2, the flow along any edge is at most $O(\log n)$ with probability at least $1 - 1/n^3$.

Hint: Use Chernoff bounds. See the theorem on relative error in the following wikipedia entry.

http://en.wikipedia.org/wiki/Chernoff_bounds

(d) Conclude that the above algorithm yields an unsplittable multicommodity flow whose value is at most an $O(\log n)$ factor less than that of an optimal unsplittable multicommodity flow, with probability at least $1 - 1/n$.