15-852 Randomized Algorithms

Homework # 6 **due: April 30, 1997**

This last homework assignment is somewhat unusual. Your job will be to look something up from the literature, digest it (figuratively), and explain it in your own words. The topic we'll be looking at is randomized algorithms for approximating dense instances of NP-hard problems. In particular, we'll be looking at a notion of approximation that only becomes interesting on dense instances — e.g., if the graph has $o(n^2)$ edges, then the problems are trivially solvable. Specifically, pick *one* of the following problems: [1]

- Given a k-colorable graph, find a partition $V_1, V_2, \ldots, V_k$ of the vertex set that is an "approximate k-coloring" in the sense that there are at most ϵn^2 violating edges (edges whose endpoints are both inside the same V_i).
- Given a graph, find a partition V_1, V_2 of the vertex set that is an "approximate MAX-CUT" in the sense that the number of edges crossing the cut is at least OPT$-\epsilon n^2$, where OPT is the value of the optimal max cut.
- Given a 3-SAT formula, find an assignment that satisfies at least OPT$-\epsilon n^3$ clauses, where OPT is the maximum number of clauses simultaneously satisfiable.

and pick one of the following papers (please select one you are not already familiar with):

- [Arora-Karger-Karpinski] STOC '95. Also available at `http://www.cs.princeton.edu/~arora/publist.html`
- [Frieze-Kannan] FOCS '96. Also available at `http://www.andrew.cmu.edu/user/af1p/regular.ps.Z`
- [Goldreich-Goldwasser-Ron] FOCS '96. Also available via `http://theory.lcs.mit.edu/~oded/ggr.html`

Your job is to describe, as cleanly as possible, an algorithm for one of these problems (and its analysis) whose running time is $poly(n) \cdot f(\frac{1}{\epsilon}, \frac{1}{\delta})$, where f is some not-necessarily-polynomial function. (These are randomized algorithms, so they will have some failure probability δ.)

[1] Or, if you prefer, you can look at the decision version of these questions. For instance, we want an algorithm that given a graph G, will output 1 if G is k-colorable and w.h.p. output 0 if G is "far from k-colorable" in the sense that you would need to remove at least ϵn^2 edges from G to make it k-colorable.