INTRO TO APPROXIMATION, CLASS 5

rounding of linear programs

EXERCISE ONE Formulate the following as integer or linear programs:

1. MINIMUM VERTEX COVER: We get an undirected graph G on input. Our goal is to find the smallest set VC such that every edge $uv \in E(G)$ has either the endpoint $u \in VC$ or $v \in VC$ (or both).
2. MINIMUM WEIGHTED DOMINATING SET: On input we get an undirected graph G with weights $w_v, w_v \geq 0$ on the vertices. The goal is to find the lightest set of vertices D such that every vertex is either in D itself, or one of its neighbors belongs to D. (Note the difference from vertex cover!)
3. MAXIMUM VERTEX CUT (MAXCUT): We have already seen this problem before. Given an undirected graph G with weights on the edges w_e, we are tasked with finding a partition of the vertices into groups V and W such that the weight of edges between each group is as high as possible.
4. 3-COLORING: A well known problem from graph theory – on input we get an undirected graph G. Design an integer program which finds a coloring of the graph using three colors such that no edge is monochromatic (sees two vertices of the same color) – and which fails if it impossible to do so.

   Hint: I do not know how to do this integer program for variables $x \in \{1, 2, 3\}$, where the numbers correspond to colors. Instead, design a clever program that has variables only in $x \in \{0, 1\}$.

EXERCISE TWO Let us practice rounding of linear programs by designing a 2-approximation algorithm for SONET RING-LOADING PROBLEM.

In this problem we have a network – a cycle with n vertices. On input we get a list of requests, each request having a source vertex and a target vertex. Every request needs to be assigned one out of two possible paths from source to target (either clockwise or counterclockwise).

Our goal is to minimize the load (total number of uses) of the most loaded edge of the cycle.

Design a integer program which models the SONET RING-LOADING PROBLEM, relax it, and then apply rounding to get a 2-approximation.

EXERCISE THREE We now consider MAX DICUT. On the input we get a directed graph $G = (V, \vec{E})$ and a non-negative weight function on the edges. Our task is to find a subset of vertices S so that $\vec{E}(S, V \setminus S)$ (the edges directed from S to the rest) have maximum possible weight.

Suggest a probabilistic $\frac{1}{4}$-approximation algorithm for MAX DICUT. This should be fairly easy, and it will not require linear programming.

EXERCISE FOUR Let us try to improve on our algorithm for MAX DICUT:

1. Suggest a natural $\{0, 1\}$-integer program solving MAX DICUT.
2. Choose each vertex v_i with probability $1/4 + x_i^*/2$, where x_i^* is the optimum of the linear relaxation of the previous integer program. Show that it is a 1/2-approximation.