INTRODUCTION TO APX – HW2

greedy algs, SAT, LP and more

Every task is worth two points. Deadline: **22. 12. 2015 17:19.** Solutions can be sent via email or handed to me in person. If some definitions are unclear or possibly incorrect, do not be afraid to send me an email.

EXERCISE ONE On the input for MAXIMUM k-CUT we get an undirected graph G and weights on the edges $w\colon E(G) \to \mathbb{R}^+$. Our goal is to partition the vertices into k disjoint sets $V_1, V_2, \ldots, V_k$ so that we minimize the sum of all "multicolored" edges (those that go from any one set to any other set).

Suggest and analyze a $\frac{k-1}{k}$-approximation algorithm for MAXIMUM k-CUT.

EXERCISE TWO You might recall the algorithm LP-SAT for the problem MAX SAT from the lecture. After combining with the naive RAND-SAT, we have produced a 3/4-approximation algorithm.

We try to avoid the combining part and apply another trick instead: we start with the optimal solution (y^*, z^*) for the exact same LP as in the algorithm LP-SAT but instead setting $P[x_i = 1] = y_i^*$ we set the probability to be $P[x_i = 1] = f(y_i^*)$, where the function $f\colon [0,1] \to [0,1]$ is defined as follows:

$$f(p) = \begin{cases} \frac{3p}{4} + \frac{1}{4} & \text{for } 0 \le p \le \frac{1}{3} \\ \frac{1}{2} & \text{for } \frac{1}{3} \le p \le \frac{2}{3} \\ \frac{3p}{4} & \text{for } \frac{2}{3} \le p \le 1 \end{cases}$$

Prove that this choice (without any sort of combination) leads to a $\frac{3}{4}$-approximation algorithm for MAX SAT.

EXERCISE THREE Consider the problem of GRAPH BALANCING, where you get an undirected graph G with weights on the edges $p\colon E(G) \to \mathbb{R}^+$. Our goal is to make the graph directed so that the most loaded vertex (the vertex with the most weight directed towards it) has minimum possible load. Formally we seek an orientation of the edges which minimizes the goal function $u = \max_{v \in V} \sum_{e \in E; e \text{ directed to } v}$.

Suggest and analyze a 2-approximation algorithm for the problem.

Tip: Linear programming might come in handy.

EXERCISE FOUR In the metric k-SUPPLIER PROBLEM we get $m + n$ points on input, where m of them are (in advance) marked as *suppliers* and the rest are *consumers*. Between all those points is a metric (with a triangle inequality as usual). Our task in this case is to select k suppliers so that we minimize the longest distance between a customer and its closest supplier.

Suggest and analyze a 3-approximation algorithm for the k-SUPPLIER PROBLEM.

Tip: This problem is related to the k-CENTER PROBLEM, which is going to be presented at the exercise session and if you miss it, you can also read about it in the Williamson, Shmoys book, chapter 2.2.