CS 49/149: Approximation Algorithms

Problem set 4. Due: 29th April, 6:59pm

General small print: Please submit all homework electronically in PDF format ideally typeset using LaTeX. You need to submit only the problems above the line. Please try to be concise – as a rule of thumb do not take more than 1 (LaTeX-ed) page for a solution. We highly encourage students to also do the problems below the line for a better understanding of the course material.

Topics in this HW: Basic Feasible Solutions, Iterative Rounding

Problem 1. Consider a feasible region of an LP $P = \{x : Ax \geq b\}$. A point $x \in P$ is called *extreme point solution* if there doesn't exist two points $y, z \in P$ and $0 < \theta < 1$ such that $x = \theta y + (1-\theta)z$. Prove that x is an extreme point solution **if and only if** it is a basic feasible solution. Don't miss the if and only if – there are two things you need to show.

Problem 2. Consider the following system of equations on a *general* (not necessarily bipartite) graph $G = (V, E)$,

$$P_M := \{0 \leq x_{uv} \leq 1, \forall (u,v) \in E : \quad x(\delta(v)) = 1, \; \forall v \in V\}$$

where $\delta(v)$ is the set of edges incident on v. Prove that in any basic feasible solution x, each entry $x_{u,v} \in \{0, 1/2, 1\}$.

Problem 3. A *hypergraph* $H = (V, E)$ is a collection of vertices V (just as in a normal graph) and a collection of *hyperedges* E where every $e \in E$ is a subset $e \subseteq V$. A k-uniform hypergraph has $|e| = k$ for all $e \in E$. (A hypergraph is a fancy way of saying set-system.)

A matching M in a hypergraph is a collection $M = \{e_1, \ldots, e_k\} \subseteq E$ such that $e_i \cap e_j = \emptyset$ for all $i \neq j$. We want to find the maximum cardinality matching in an arbitrary k-uniform hypergraph. Design and analyze a $(k - 1 + 1/k)$-factor approximation algorithm for this problem. You may proceed in the following manner.

- Write an LP relaxation with a variable x_e for each edge e.
- Prove that there exists some edge e with x_e larger than some quantity.
- Using this find an iterated rounding approximation algorithm.

Problem 4. Suppose we are given a *real valued* matrix $A \in \Re^{m \times n}$ and a vector $x \in [0,1]^n$ satisfying $Ax = b$ for some $b \in \Re^m$. Also suppose for each column of A, the sum of the positive entries is $\leq \Delta$ and the sum of the negative entries is $\geq -\Delta$ for some $\Delta > 0$. Find a $\{0,1\}$-vector z such that $(Az)_i \leq b_i + \Delta$ for $i \in [m]$. For partial credit prove this when A is a non-negative matrix.

Problem 5. In the Tree Augmentation Problem, suppose all links were "up-links", that is, each $f = (u, v)$, the node v was an ancestor of u. Prove that the LP-relaxation for this case is integral.

Also design a polynomial time dynamic programming based combinatorial exact algorithm for this problem.