CS270 Combinatorial Algorithms and Data Structures — **Last Updated:** April 2, 2015
Homework 2 — **Due Date: April 16, 2015**

You are allowed to discuss with others but not allowed to use any references except the course notes. Please list your collaborators for each question. This will not affect your marks. In any case, you must write your own solutions.

There are totally 90 marks, and the full mark is 80. This homework is counted 16% of the course.

Some hints will be provided on April 9 in a separate file.

---

1. **Extreme Point Solutions**

   (10 marks) We consider linear programming in the inequality form: $\min\langle c, x\rangle, Ax \leq b$. In L12, we give the definitions of (1) vertex solutions (2) extreme point solutions and (3) basic solutions, and proved that (1) and (3) are equivalent. In this question, you are asked to prove that they are all equivalent by proving that (3) implies (2) and (2) implies (1).

2. **Minimum s-t Cut**

   (10 marks) Prove that the LP for minimum s-t cut as shown in L15 is integral.

$$\begin{aligned} \min \sum_e c_e d_e & & \\ d_{uv} + y_v - y_u &\geq 0 & \forall\, uv \in E \\ y_s - y_t &\geq 1 & \\ y_v &\geq 0 & \forall\, v \in V \\ d_e &\geq 0 & \forall\, e \in E \end{aligned}$$

3. **Balanced Coloring**

   (10 marks) Given a hypergraph $G = (V, E)$ where each hyperedge $e \in E$ is a subset of V, our goal is to color the vertices of G using $\{-1, +1\}$ such that each hyperedge is as balanced as possible. Formally, given a coloring $\psi : V \to \{-1, +1\}$ on the vertices, we define $\Delta(e) = \sum_{v \in e} \psi(v)$ and $\Delta(G) = \max_{e \in E} |\Delta(e)|$. Prove that if the maximum degree of the hypergraph is d (i.e. each vertex appears in at most d hyperedges), then there is a coloring with $\Delta(G) \leq 2d - 1$.

   You may find it useful to consider the following LP, where initially we set $B_e = 0$ for all $e \in E$.

$$\begin{aligned} \sum_{v \in e} x_v &= B_e & \forall\, e \in E \\ -1 \leq x_v &\leq 1 & \forall\, v \in V \end{aligned}$$

4. **Cooperative Game Theory**

   (10 marks) Consider a set $N = \{1, \ldots, n\}$ of n players. Let $v : 2^N \to \mathbb{R}_+$ be a value function, i.e. for $S \subseteq N$, the value $v(S)$ is the total worth that the members of S can earn without any help from the players outside S. By default, we set $v(\emptyset) = 0$. Naturally, the total worth $v(N)$ is shared among all the players, and we would like to know how to share it so that no subset S has the incentive to deviate and obtain an outcome that is better for all of its members. Specifically, consider an allocation vector $x \in \mathbb{R}^n$ where x_i represents the payoff to player i for $1 \leq i \leq n$. We say that a subset S can improve upon an allocation if and only if $v(S) > x(S) = \sum_{i \in S} x_i$. We say that an allocation x is stable if

   $$x(N) = v(N), \quad x(S) \geq v(S) \quad \forall S \subseteq N.$$

   Use LP duality to give a necessary and sufficient condition for the existence of a stable allocation.

5. **Multiplicative Weights Update Method**

   (10 marks) Consider the maximum flow problem from s to t on a directed graph, where each edge has capacity one. A fractional s-t flow solution with value k is an assignment of each edge e to a fractional value $x(e)$, satisfying that

   $$\sum_{e \in \delta^{\text{out}}(s)} x(e) = \sum_{e \in \delta^{\text{in}}(t)} x(e) = k,$$

   $$\sum_{e \in \delta^{\text{in}}(v)} x(e) = \sum_{e \in \delta^{\text{out}}(v)} x(e) \quad \forall v \in V - \{s, t\}, \text{ and}$$

   $$0 \leq x(e) \leq 1 \quad \forall e \in E.$$

   Use the multiplicative weights update method to solve this LP by reducing the flow problem to the problem of finding shortest paths between s and t. Analyze the convergence rate and the total complexity of your algorithm to compute a flow of value $k(1 - \epsilon)$ for any $\epsilon > 0$.

6. **Sums of Squares and Minimization**

   (10 marks)

   (a) Let $p(x) \in \mathbb{R}[x]$ be a univariate polynomial of degree d with real coefficients. We would like to decide whether $p(x)$ is a *sum of squares*, i.e., if it can be written as $p(x) = q_1(x)^2 + \cdots + q_m(x)^2$ for some $q_1(x), \ldots, q_m(x) \in \mathbb{R}[x]$. Formulate this problem as the feasibility of a semidefinite program.

   (b) Let us call a polynomial $p(x) \in \mathbb{R}[x]$ *nonnegative* if $p(x) \geq 0$ for all $x \in \mathbb{R}$. Obviously, a sum of squares is nonnegative. Prove that the converse holds as well: Every nonnegative univariate polynomial is a sum of squares.

   (c) Let $p(x) \in \mathbb{R}[x]$ be a given polynomial. Express its global minimum $\min\{p(t) : t \in \mathbb{R}\}$ as the optimum of a suitable semidefinite program.

7. **Minimum Multicut**

   (15 marks) We are given a graph $G = (V, E)$ and k pairs of source-sink vertices, $s_i, t_i \in V$ for $i = 1, \ldots, k$. We wish to find a subset of edges $F \subseteq E$ that minimizes $|F|$ such that for each $i = 1, \ldots, k$, there is no s_i-t_i path in $(V, E - F)$.

Consider the following vector program:

$$\begin{array}{lrll} \text{minimize} & \sum_{(i,j)\in E}(1 - v_i \cdot v_j) & & \\ \text{subject to} & v_{s_i} \cdot v_{t_i} & = 0, & i = 1, \ldots, k, \\ & v_j \cdot v_j & = 1, & \forall j \in V, \\ & v_j & \in \mathbb{R}^n, & \forall j \in V. \end{array}$$

Consider the demand graph $H = (V, E')$, where $E' = \{(s_i, t_i) : i = 1, \ldots, k\}$. Let Δ be the maximum degree of a vertex in the demand graph. Suppose that the optimal value of the vector programming relaxation is $\epsilon|E|$. Consider the following algorithm. We draw $t = \lceil \log_2(\Delta/\epsilon) \rceil$ random unit vectors $r_1, \ldots, r_t$. The t random vectors define 2^t different regions into which the vectors v_i can fall: one region for each distinct possibility of whether $r_j \cdot v_i \geq 0$ or $r_j \cdot v_i < 0$ for all $j = 1, \ldots, t$. Remove all edges (i, j) from the graph such that v_i and v_j are in different regions. If for any s_i-t_i pair, there still exists an s_i-t_i path, remove all edges incident on s_i. We now analyze this algorithm.

(a) Prove that the vector program is a relaxation of the minimum multicut problem.

(b) For any $(i, j) \in E$, prove that the probability that i and j are in different regions is at most $t \cdot \sqrt{1 - v_i \cdot v_j}$.

(c) Prove that for any $i = 1, \ldots, k$, the probability that we end up removing all the edges incident on s_i is at most $\Delta 2^{-t}$.

(d) Show that the expected number of edges removed is at most $O(\sqrt{\epsilon} \log(\Delta/\epsilon)|E|)$.

For the final item, it may be useful to use Jensen's Inequality, which states that for any convex function f (that is, $f''(x) \geq 0$) and any positive p_i,

$$f\left(\frac{\sum_i p_i x_i}{\sum_i p_i}\right) \leq \frac{1}{\sum_i p_i} \sum_i p_i f(x_i).$$

8. **Theta Function**

(15 marks) Consider the Theta function for the maximum independent set problem on a graph $G = (V = [1, n], E)$.

$$\begin{array}{rcll} \vartheta(G) & := & \max \sum_{i \in V} \|v_i\|^2 & \\ \langle v_i, v_j \rangle & = & 0 & \forall ij \in E \\ \langle v_0, v_i \rangle & = & \langle v_i, v_i \rangle & \forall i \in V \\ \langle v_0, v_0 \rangle & = & 1 & \end{array}$$

(a) Construct the dual program for the Theta function and write it as a vector program.

(b) Show that it gives a feasible solution to the following vector clique cover problem with objective value $\vartheta(G)$. (This is the SDP for graph coloring for the complement graph $\overline{G}$.)

$$\begin{array}{rcll} \min\ t & & & \\ \langle u_i, u_j \rangle & \leq & \dfrac{-1}{t-1} & \forall ij \notin E \\ \langle u_i, u_i \rangle & = & 1 & \forall i \in V \end{array}$$