CS 224 Advanced Algorithms — Fall 2014

Problem Set 6

Due: 11:59pm, Wednesday, November 5th
Submit to: `cs224-f14-assignments@seas.harvard.edu`
Solution maximum page limit: 5 pages

See homework policy at `http://people.seas.harvard.edu/~minilek/cs224/hmwk.html`

Problem 1: Given an unweighted graph $G = (V, E)$ where $|V| = n, |E| = m$, consider $\mathcal{M} \subset \mathbb{R}^m$ defined by constraints $\sum_{e \ni v} x_e = 1 \ \forall v \in V$, and $x_e \in \{0, 1\} \ \forall e \in E$. Clearly any $x \in \mathcal{M}$ represents the incidence vector of a perfect matching in G. (Recall that a matching M is a subset of E such that no two edges in M share a common vertex, and M is *perfect* if every vertex is incident upon exactly one edge in M.) In the minimum cost perfect matching problem we would like to compute $x \in \mathcal{M}$ which minimizes $c^T x$ for some $c \in \mathbb{R}^m, c \geq 0$. Unfortunately this is not linear programming due to integrality constraints.

We could relax the problem to a linear program by defining $P_{\mathcal{M}} = \{x \in \mathbb{R}^m : \sum_{e \ni v} x_e = 1, \ \forall x \in V; x \geq 0\}$. Then we could minimize $c^T x$ subject to $x \in P$. This question investigates when solving this LP relaxation actually gives an integral solution x^*.

(a) (4 points) (In this problem part, n, m are not related to $|V|, |E|$ above.) For a matrix $A \in \mathbb{R}^{m \times n}$ and vector $b \in \mathbb{R}^m$ with integer entries, $n \geq m$, consider a polytope $P \subset \mathbb{R}^m$ defined by $\{x : Ax = b; x \geq 0\}$. We say A is *totally unimodular (TUM)* if every square submatrix of A has determinant in the set $\{-1, 0, 1\}$ (by "submatrix", we mean any matrix formed by taking an arbitrary subset of rows and arbitrary subset of columns). Show if A is TUM then every vertex of P is integral (that is, it has all integer entries). **Hint:** Cramer's rule.

In the following two parts, $A \in \mathbb{R}^{n \times m}$ is the constraint matrix defining $P_{\mathcal{M}}$ above.

(b) (4 points) Show that if G is bipartite, then A is TUM. **Hint:** Use induction on the size of the submatrix.

(c) (2 points) Show that if A is TUM then G is bipartite.

Note that by 1(a) and 1(b), we can solve the minimum cost perfect matching problem in bipartite graphs using any off-the-shelf linear programming solver which guarantees a vertex as its output (though there are more efficient combinatorial algorithms).

Problem 2: (10 points) Consider the following problem. We are given a directed graph $G = (V, E)$ and a "root" vertex $r \in V$ identified as the root. Furthermore each edge e has a cost $c_e > 0$. We would like to find a subgraph $T \subseteq E$ such that, when viewed as an undirected graph, T is a spanning tree of G. When viewed with edge directions however, we also require that all edges in T point away from the root r. Amongst all such T, we would like the cost $c(T) = \sum_{e \in T} c_e$ to be minimized. We call this the *cheapest rooted tree problem.*

Consider the integer program

$$\begin{aligned} \min \sum_{e \in E} c_e x_e & & & \\ s.t. \quad \sum_{\substack{e=(u,v)\in E \\ u \notin S, v \in S}} x_e & \geq & 1, \ \forall S \subset V \backslash \{r\} \\ x_e & \in & \{0,1\}, \ \forall e \in E \end{aligned}$$

(a) (4 points) Show that given an optimal solution to the above integer program, we can obtain an optimal solution to the cheapest rooted tree problem in time $poly(|V|, |E|)$.

(b) (6 points) Suppose we relax the constraints $x_e \in \{0, 1\}$ to $x_e \geq 0$. Show the corresponding linear program can be solved in polynomial time using the ellipsoid algorithm, despite there being exponentially many constraints. **Hint:** given an infeasible point in an ellipsoid iteration, find a separating hyperplane using minimum cut computations.

Problem 3: (10 points) In the *least squares regression* problem, we are given $A \in \mathbb{R}^{n \times d}$, $n \geq d$, and a vector $b \in \mathbb{R}^n$. The goal is to compute x^* which minimizes $\|Ax - b\|_2^2$. It is well known that if A is of full column rank, then $x^* = (A^T A)^{-1} A^T b$. In what follows, the *condition number* $\kappa(A)$ is the largest singular value of A divided by the smallest singular value. Recall the singular values of A are the square roots of the eigenvalues of $A^T A$.

(a) (3 points) Prove that $x^* = (A^T A)^{-1} A^T b$.

(b) (7 points) Suppose we have some $\tilde{x}$ such that $\|A\tilde{x} - b\|_2^2 \leq \beta \cdot \|Ax^* - b\|_2^2 = \beta \cdot \mathsf{OPT}$ for some $\beta > 1$, and we know β. How would you efficiently obtain some $\tilde{x}'$ such that $\|A\tilde{x}' - b\|_2^2 \leq (1 + \varepsilon)\mathsf{OPT}$? State the steps in your algorithm precisely (given your algorithm description, someone should be able to code it up in a few minutes without much thought — assume the programmer is given the largest and smallest singular values of A as input). Give some bounds on your running time.

Problem 4: (1 point) How much time did you spend on this problem set? If you can remember the breakdown, please report this per problem. (sum of time spent solving problem and typing up your solution)