CSCI3160: Regular Exercise Set 13

Prepared by Yufei Tao

Problem 1 (Reduction from Hitting Set to Set Cover). Given an instance to the hitting set problem, explain how to convert it to a set cover problem.

Problem 2 (Reduction from Set Cover to Hitting Set). Given an instance to the set cover problem, explain how to convert it to a hitting set problem.

Problem 3. In the hitting set problem, we are given a collection of sets $\mathcal{S}$, where each set $S \in \mathcal{S}$ is a subset of some universe U. We want to find a hitting set $H \subseteq U$ of the smallest size (recall that H is an hitting set if $H \cap S \neq \emptyset$ for every $S \in \mathcal{S}$). Let OPT be the size of an optimal hitting set. Design a polynomial time algorithm that returns a hitting set of size at most $\text{OPT} \cdot (1 + \ln |\mathcal{S}|)$.

Problem 4. Let $G = (V, E)$ be an undirected simple graph where each edge $e \in E$ is associated with a non-negative weight $w(e)$. For any vertices $u, v \in V$, define $spdist(u, v)$ as the shortest path distance between u and v. Given a subset $C \subseteq V$, define its *cost* as

$$cost(C) \quad = \quad \max_{u \in V} \min_{c \in C} spdist(c, u).$$

Fix an integer $k \in [1, |V|]$. Let OPT be the smallest cost of all subsets $C \subseteq V$ with $|C| = k$. Design an algorithm to find a size-k subset with cost at most $2 \cdot \text{OPT}$. Your algorithm must run in time polynomial to $|V|$.

Problem 5. Consider the k-center problem on a set P of n 2D points. Our lecture made the assumption that the Euclidean distance of any two points can be computed precisely in polynomial time. This is not a realistic assumption (because the computation requires calculating square roots). Modify our 2-approximate algorithm to make it run in polynomial time without the assumption.

Problem 6.** Let P be a set of n 2D points. Given a subset $C \subseteq P$, define:

- (for each point $p \in P$) $dist_C(p) = \min_{c \in C} dist(c, p)$, where $dist(c, p)$ represents the Euclidean distance between c and p;
- $cost(C) = \max_{p \in P} dist_C(p)$.

Fix a real value $r > 0$. Call a subset $C \subseteq P$ an *r-feasible subset* if $cost(C) \leq r$. Prove: unless P = NP, there does not exist an algorithm that can find an r-feasible subset with the smallest size in time polynomial to n. You can assume that the Euclidean distance of any two points can be computed in polynomial time.

(Hint: Show that the existence of such an algorithm implies a polynomial time algorithm for the k-center problem.)