CS 598TMC, Fall 2018
Timothy Chan

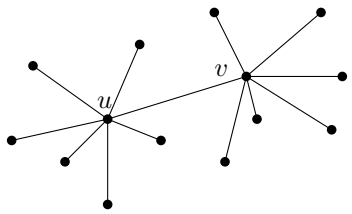
Homework 1 (due Sep 28 Friday (11am in class))

Instructions: You may work in groups of at most 2. Hand in one set of solutions per group. Acknowledge any discussions you have with other students and other sources you have consulted. Solutions must be written *in your own words*.

1. [*22 pts*] Consider the following variant of the minimum spanning tree problem: Given a set P of n points in $\mathbb{R}^2$, find a spanning tree T that minimizes $c(T) := \max_{p,q \in P} d_T(p,q)$, where $d_T(p,q)$ denotes the Euclidean length of the path from p to q in T. (The motivation comes from designing networks to minimize maximum "delay".)

   (a) [*7 pts*] Prove that in the optimal tree T, there exists an edge uv in T such that all vertices of $P - \{u, v\}$ are adjacent to u or v.

   (Hint: consider the longest path in T. Pick some edge uv in the "middle" of this path, and re-link every vertex to either u to v, and argue that the cost decreases.)



   (b) [*5 pts*] Using (a), give a polynomial-time exact algorithm.

   (c) [*10 pts*] Using (a) and (b), give a $(1+\varepsilon)$-factor approximation algorithm with running time $O(n + 1/\varepsilon^c)$ for some constant c.

2. [*23 pts*] In this question, we will analyze a different algorithm for approximating the width of a point set P in two dimensions. We use an *iterative* approach, which builds up a coreset incrementally by adding two new points in every iteration.

   Let Δ^* denote the diameter of P. Let $w^* = \text{wid}(P)$ denote the width of P. Let $w_v(P) = \max_{p,q \in P}(p - q) \cdot v$ denote the width of P along direction v. (Recall that $\text{wid}(P)$ is the minimum of $w_v(P)$ over all unit vectors v.) Here is the proposed algorithm:

   1. let s be any point in P, and let t be the farthest point in P from s
   2. $S_0 = \{s, t\}$
   3. for $j = 0, 1, \ldots$ do
   4. compute $\text{wid}(S_j)$, and let v_j be the unit vector with $\text{wid}(S_j) = w_{v_j}(S_j)$
   5. let p_j, q_j be the points of P with $w_{v_j}(P) = (p_j - q_j) \cdot v_j$
   6. if $\text{wid}(S_j) \geq (1-\varepsilon) w_{v_j}(P)$ then return $\text{wid}(S_j)$
   7. $S_{j+1} = S_j \cup \{p_j, q_j\}$

(a) [*2 pts*] First show that when the algorithm terminates, it indeed returns a $(1 + O(\varepsilon))$-approximation of the width of P.

(b) [*3 pts*] Show that if $\angle(v_i, v_j) = \theta$, then $u \cdot v_i$ and $u \cdot v_j$ differ by at most $O(\theta\Delta^*)$ for any vector u of length at most Δ^*.

(c) [*7 pts*] Using (b) (twice), show that for every $i < j$ before the last iteration, we must have $\angle(v_i, v_j) = \Omega(\varepsilon w^*/\Delta^*)$.

(d) [*3 pts*] Prove that for every v, if $\angle(v_0, v) = \theta$, then $w_v(\{s, t\}) = \Omega(\theta\Delta^*)$.

(e) [*3 pts*] Using (d), show that for every j, we must have $\angle(v_0, v_j) = O(w^*/\Delta^*)$.

(f) [*5 pts*] Using (c) and (e), conclude that the number of iterations is at most $O(1/\varepsilon)$ and the algorithm runs in $O((1/\varepsilon^c)n)$ time for some constant c.

3. [*10 pts*] Consider the following "special-parabola fitting" problem: Given a set P of n points in $\mathbb{R}^2$, find real numbers α, β that minimize $\max_{(x,y)\in P} |(\alpha x + \beta)^2 - y|$.

   Describe a $(1+\varepsilon)$-approximation algorithm with running time $O(n+1/\varepsilon^c)$ for some constant c.