Solutions to Ex. 4

Yinon Nahum

July 18, 2018

1 Question 1

The algorithm is incorrect. (P1) holds trivially since the weak diameter of each ball is at most $2\delta \leq D/2$, but (P2) does not. A counter example is the line graph where the nodes are ordered $1, 2, \ldots, n$ from left to right, π is the identity order, i.e., $\pi(i) = i$ for every i, and $0.5n \leq D \leq n$.
Note that $V_1 = \{1, \ldots, \delta\}$ and the nodes $\delta+1, \delta+2, \ldots, n$ are each in a separate component. In particular, since $\delta \leq D/4 < n$, we have that n and $n-1$ are separated with probability 1, but $\alpha \cdot dist_G(n-1, n) = \alpha = o(1)$.
We correct this by choosing π to be a uniformly random permutation. Fix nodes u and v and index i. We say that $z_i = \pi(i)$ *separates* if it is the first to include exactly one of u and v in its ball. WLOG, assume z_i is closer to v than to u.

$$\Pr[z_i \text{ separates}] = \Pr[z_i \text{ separates} \mid dist(z_i, v) \leq \delta < dist(z_i, u)] \cdot \Pr[dist(z_i, v) \leq \delta < dist(z_i, u)] \,. \tag{1}$$

Note that if $dist(z_i, v) > \delta$ or $dist(z_i, u) \leq \delta$, then z_i cannot separate, thus the case is omitted. Due to the choice of δ and the triangle inequality,

$$\Pr[dist(z_i, v) \leq \delta < dist(z_i, u)] = \frac{dist(z_i, u) - dist(z_i, v)}{D/4 - D/8 + 1} \leq \frac{dist(v, u)}{D/8} \,. \tag{2}$$

If z_i separates, then $dist(z_i, \{v, u\}) < dist(z_j, \{v, u\})$ for every $j < i$, where the distance to a set is defined in q. 5. Therefore,

$$\begin{aligned}
&\Pr[z_i \text{ separates} \mid dist(z_i, v) \leq \delta < dist(z_i, u)] \\
&\quad \leq \Pr[\forall j < i : dist(z_i, \{v, u\}) < dist(z_j, \{v, u\}) \mid dist(z_i, v) \leq \delta < dist(z_i, u)] \\
&\quad = \Pr[\forall j < i : dist(z_i, \{v, u\}) < dist(z_j, \{v, u\})] = 1/i \,,
\end{aligned}$$

where the penultimate equality follows since the unconditional event depends only on π and the event conditioned on depends only on δ, and the last equality follows from the choice of π. Plugging this and Ineq. (2) into Eq. (1), we have

$$\Pr[z_i \text{ separates}] \;\leq\; \frac{1}{i} \cdot \frac{8dist(v,u)}{D} \;.$$

Taking union bound over all i yields (P2).

Common mistake: *arbitrary permutation means the worst permutation, thus the algorithm fails. Some of you mistook arbitrary to mean that you can choose the permutation and claimed that the algorithms succeeds.*

2 Question 2

(Q1) - By the definition of r_v we have

$$C(E(v,r_v)) > (1+\alpha)C(E(v,r_v-1)) > \cdots > (1+\alpha)^{r_v-1}C(E(v,1)) \;.$$

Since $C(E(v,1)) \geq 1$ and $C(E(v,r_v)) \leq C$, we have

$$C \;>\; (1+\alpha)^{r_v-1} \;=\; \left((1+\alpha)^{1/\alpha}\right)^{\alpha(r_v-1)} \;\geq\; 2^{\alpha(r_v-1)} \;,$$

where the last inequality is Bernoulli's. Therefore,

$$r_v < 1 + \frac{\ln C}{\alpha \ln 2} = 1 + \frac{D}{4\ln 2} \leq D/2 \;.$$

Therefore, the diameter of each ball is at most D.

Note: *Using Bernoulli's inequality, we assumed $\alpha \leq 1$. For large α's, the algorithm might fail. Since this was not mentioned in the question, points were not deducted for certain false arguments.*

(Q2) - Note that there are at most $C(E(v_\ell, r_{v_\ell}+1)) - C(E(v_\ell, r_{v_\ell})) \leq \alpha \cdot C(E(v_\ell, r_{v_\ell}))$ inter-cluster edges created at iteration ℓ by the cluster centered at v_ℓ. Summing over all iterations, the total number of inter-cluster edges is at most $\alpha \sum_\ell C(E(v_\ell, r_{v_\ell})) \leq \alpha C(E)$.

3 Question 3

(a) - We find a tree with average stretch at most 7/3. Numbering the nodes on the ring clockwise from 1 to n, we take all $n/2-1$ edges on the path $(1,2,3,\ldots,n/2)$ and all $n/2$ chords. The stretch of all $(n-1)$ selected edges is 1. The stretch of the $n/2-1$ edges $\{i,j\} \subseteq [n/2+1, n]$ is 3 due to the path $(i, i-n/2, j-n/2, j)$. The stretch of the edge

$\{1, n\}$ is $n/2$ due to the path $(1, 2, \ldots, n/2, n)$. Symmetrically, for the edge $\{n/2, n/2+1\}$ the stretch is also $n/2$. Thus, the total stretch is

$$(n-1) \cdot 1 + (n/2 - 1) \cdot 3 + 2 \cdot n/2 \ \leq \ 7n/2 \ .$$

Since W_n has $3n/2$ edges, the average stretch is at most 7/3. Note that the selected edged form a tree since there are $n-1$ such and the stretch of each original edge is finite.
Note: *choosing* $n/2$ *cycle edges and* $n/2 - 1$ *chords results in average stretch 8/3.*

(b) - If the grid consists of 1 node, return the empty tree. Otherwise, we divide the grid to 4 quarters, solve recursively to obtain trees T_1, T_2, T_3, T_4, each from one quarter, and connect them by adding 3 edges forming a path of length 3 connecting all 4 trees in the center of the original grid. Then return the resulting tree.
We first note that the diameter of the tree is $O(\sqrt{n})$. To show this, let $D(n)$ denote the diameter of the returned tree. Then $D(n) \leq 2D(n/4) + 3$, since we can walk from $v \in T_i$ to $u \in T_j$ by first walking from v to the 3-path using edges in T_i, then walking on the 3-path to reach T_j, and then walking to u using edges in T_j. By the Master Theorem, this yields $D(n) = O(\sqrt{n})$.
We then note that the total stretch is $O(n \log n)$. To show this, let $T(n)$ denote the total stretch of the returned tree. Then $T(n) \leq 4T(n/4) + O(n)$, since the stretch of an edge in each quarter is $T(n/4)$ and the stretch of the $2\sqrt{n}$ inter-quarter edges is $O(\sqrt{n})$ as this is the diameter. By the Master Theorem, this yields $T(n) = O(n \log n)$. Since an n-vertex grid contains $\Omega(n)$ edges, the average stretch is $O(\log n)$.
Common mistakes: *not connecting* T_1, T_2, T_3, T_4 *properly, resulting in a large tree diameter; struggling with complicated inductive proofs, instead of using the Master Theorem; not showing that the tree diameter is* $O(\sqrt{n})$.

4 Question 4

Replace LDD with DetDecomp. We saw in class (Claim 6.4 in the lecture notes), $d_T(u,v) \geq d_G(u,v)$. To prove the upper bound, consider a call to *DetDecomp*$(G', D/2^{i-1})$ occuring on level i of the tree, and denote $E' = E(G')$. By the Lemma in q. 2, this call creates at most $\frac{4\ln(|E'|)}{D/2^{i-1}} \cdot |E'|$ inter-cluster edges, and by what we saw in class (Claim 6.5 in the lecture notes), each has a stretch of at most $4D/2^{i-1}$. This leads to a total stretch of at most

$$\frac{4\ln(|E'|)}{D/2^{i-1}} \cdot |E'| \cdot \frac{4D}{2^{i-1}} \ = \ 16|E'| \ln |E'| = O(|E'| \log n) \ .$$

Since each edge is cut at some call to *DetDecomp*, Summing over all calls in level i, we obtain a total stretch of $O(|E| \log n)$. Summing over all the levels, we obtain a total stretch of $O(|E| \cdot \log n \cdot \log(Diam(G)))$. Thus, the average stretch is $O(\log n \cdot \log(Diam(G)))$, as required.

Common mistakes: *forgetting that level i has many calls to DetDecomp; forgetting that the parameters D and α, given to DetDecomp, depend on the level in the tree; trying to prove claims similar to the ones seen in class rather than citing them*

5 Question 5

The algorithm creates an $\alpha-$stretch spanning tree subgraph T, then runs a dynamic programing algorithm on T and returns the resulting set C. Since $d_T(v,u) \geq d_G(v,u)$ for every v and u, we have

$$\mathbb{E}\left[\sum_v d_G(v,C)\right] \leq \mathbb{E}\left[\sum_v d_T(v,C)\right] \leq \mathbb{E}\left[\sum_v d_T(v,C^*)\right] = \sum_v \mathbb{E}\left[d_T(v,C^*)\right] ,$$

where the second inequality follows from the optimality of the dynamic programming algorithm. Letting u^* be the vertex satisfying $\mathbb{E}\left[d_T(v,u^*)\right] = \min_{u\in C^*} \mathbb{E}\left[d_T(v,u)\right]$, we have

$$\mathbb{E}\left[\sum_v d_G(v,C)\right] \leq \sum_v \mathbb{E}\left[d_T(v,u^*)\right] = \sum_v \min_{u\in C^*} \mathbb{E}\left[d_T(v,u)\right] \leq \alpha \cdot \min_{u\in C^*} d_G(v,u) = \alpha \cdot OPT.$$

This algorithm requires that T be a subgraph of G. However, if we require 2α approximation, it would suffice to have $V(G) \subseteq V(T)$, since we can replace any node $u \in C \setminus V(G)$ by $z \in V(G)$, the closest node to u in T. For every $v \in V(G)$ we have $d_T(v,z) \leq d_T(v,u) + d_T(u,z) \leq 2d_T(v,u)$, yielding a 2-approximation to C, and 2α approximation to OPT. While the latter was intended, both answers were accepted.

Common mistake: *most of you did not show why* $\mathbb{E}\left[d_T(v,C^*)\right] \leq \min_{u\in C^*} \mathbb{E}\left[d_T(v,u)\right]$. *Note that this inequality holds since the minimum function is concave. In fact, if the distance to a set were defined as the distance to the farthest vertex in the set, the inequality would be reversed, i.e., we would have* $\mathbb{E}\left[d_T(v,C^*)\right] \geq \max_{u\in C^*} \mathbb{E}\left[d_T(v,u)\right]$.

Dynamic Programming: *The required dynamic programming algorithm was much more complicated than anticipated (such algorithm can be found in [1]). Thus, it was given no weight in the grade. However, any worthy attempt to provide an algorithm was rewarded with 5 points bonus.*

References

[1] Arie Tamir. An o (pn2) algorithm for the p-median and related problems on tree graphs. *Operations Research Letters*, 19(2):59–64, 1996.