Advanced Algorithms 2024 18.11, 2024

Sample Solutions 08

Lecturer: Maximilian Probst *Teaching Assistant: Patryk Morawski*

1 A Near-Linear Time Algorithm for Spanners

1. We will split the edges we add into two categories: the edges we add during one of the $k-1$ main phases of the algorithm and the edges added in the last phase. We start with the edges in each of the main phases.

   Claim 1. *The expected number of edges added in each phase is at most $n^{1+1/k}$.*

   Proof. Let v be a vertex with d neighboring clusters at the begining of the phase. Then, v adds at most d new edges to our spanner if both the cluster of v and all of the neighboring clusters of v are marked as dead. Otherwise, v adds at most 1 vertex. Thus,

   $$\mathbb{E}[\#\text{ edges added by } v] \leq (1-n^{-1/k})^d d + 1 \leq e^{-d \cdot n^{-1/k}} d + 1 = \mathcal{O}(n^{1/k})$$

   and by the linearity of expectation

   $$\mathbb{E}[\#\text{ edges added}] \leq n \cdot \mathcal{O}(n^{1/k}) = \mathcal{O}(n^{1+1/k}).$$

   □

   By the claim, during the $k-1$ main phases we add $\mathcal{O}(n^{1+1/k}$ edges in expectation. It thus remains to show that we do not add too many edges in the last phase.

   Claim 2. *During the last phase of the algorithm, we add $\mathcal{O}(n^{1+1/k})$ edges in expectation.*

   Proof. Let X be the number of surviving clusters after the $k-1$ main phases. Note that in the last phase we add at most $n \cdot X$ new edges, since every vertex adds at most one edges for a given cluster. Thus,

   $$\mathbb{E}[\#\text{ edges added in the last phase}] \leq \mathbb{E}[n \cdot X] = n\mathbb{E}[X].$$

   Now, a given cluster survives each phase with probability $n^{-1/k}$. In particular, the probability that this cluster survives all of the $k-1$ main phases is $n^{-(1-1/k)}$. By linearity of expectation, since we started with n clusters, the expected number of surving clusters after phase $k-1$ is $n \cdot n^{-(1-1/k)} = n^{1/k}$. Thus, we get

   $$\mathbb{E}[\#\text{ edges added in the last phase}] \leq n\mathbb{E}[X] = \mathcal{O}(n^{1+1/k})$$

   □

   The two claims together show that we in expectation add $\mathcal{O}(n^{1+1/k})$ edges to our spanner.

   Note that to obtain an algorithm that returns a spanner with $\mathcal{O}(n^{1+1/k})$ edges with high probability, we could simply repeat this algorithm $\log n$ times and take the spanner with minimum number of edges.

2. We first prove that in the beginning of each phase, the vertices that are in the same cluster are not far away from each other in H. More specifically we prove the following lemma.

   Claim 3. *For any cluster $\mathcal{C}$ at the beginning of phase i, the distance (in H) of the root of this cluster and any other vertex in this cluster is at most $i-1$. This implies that the distance between any two vertices in this cluster is at most $2(i-1)$.*

   Proof. We prove this statement by induction. For $i = 1$, each cluster is just a single vertex so the statement holds. For $i > 1$, we now know that at the start of phase $i-1$ the distance between any vertex in $\mathcal{C}$ and the root was at most $(i-1)-1 = i-2$. Later in the phase $i-1$ some new vertices might join our cluster. However, for every such vertex v we add an edge to an existing vertex in the cluster. In particular, the distance from v to the root of our cluster is now at most $i-2+1 = i-1$. This shows the first part of the statement.

   For the second part, notice that for any two vertices u, v in $\mathcal{C}$ we have

   $$dist_H(u,v) \leq dist_H(u,r) + dist_H(r,v) \leq 2(i-1)$$

   by the triangle inequality, where r is the root of $\mathcal{C}$. □

   Using the above claim, we will now argue that if uv is an edge in G, then the distance between u and v in H at the end of the algorithm cannot be large.

   Claim 4. *Let $uv \in E(G)$. Then, after the last phase of the algorithm, $dist_H(u,v) \leq 2k-1$.*

   Proof. Suppose first in some main phase t of the algorithm, both u and v appeared in the same cluster $\mathcal{C}$. Then, by Claim 3, at the beginning of phase $t+1$, the distance between u and v in H is at most $2(t-1) < 2k-1$. Since we only add edges to H afterwards, the distance at the end of the algorithm can only increase, so the claim holds.

   Now suppose that u and v never belong to the same cluster during an execution of the algorithm. Then, there is some phase (possibly the last one) where both u and v die at the same time. But then, by definition, we add an edge from u to some vertex v' in the cluster of v. In particular, again by Claim 3 and by the triangle inequality we get

   $$dist_H(u,v) \leq 1 + dist_H(v',v) \leq 1 + (2k-2) = 2k-1.$$

   □

   We now show that Claim 4 implies that we get a $(2k-1)$-spanner. Indeed, let $u, v \in V(G)$ with $dist_G(u,v) = d$ and let $u = w_0, w_1, \ldots, w_{d-1}, w_d = v$ be a shortest path between u and v in G. By the triangle inequality and the fact that each $w_i w_{i+1}$ is an edge in G, we get

   $$dist_H(u,v) \leq \sum_{i=0}^{d-1} dist_H(w_i, w_{i+1}) \leq \sum_{i=0}^{d-1} 2k-1 = d \cdot (2k-1).$$

3. A single phase of the algorithm can clearly be implemented in time $\mathcal{O}(m)$. Indeed, marking the vertices as dead or surviving can be implemented in linear time. Then, scanning the neighborhood of a vertex takes time proportional to its degree, and thus for all vertices this will again give $\mathcal{O}(m)$ time. Since we have $\tilde{\mathcal{O}}(1)$ phases of our algorithm, we get the total runtime of $\tilde{\mathcal{O}}(m)$.

2 Very Sparse Spanners

1. Let m be the number of edges in H. For a vertex $v \in V(H)$ let $B_v^{(i)} \subseteq V(H)$ denote the set of vertices with distance at most i to v. Let now $u, v \in V(H)$ be arbitrary and assume for contradiction that $dist_H(u,v) > d = 20\phi^{-1}\log m$. Then, it must hold that $B_v^{(d/2)} \cap B_u^{(d/2)} = \emptyset$. Indeed, if $w \in B_v^{(d/2)} \cap B_u^{(d/2)}$, then $dist_H(u,v) \leq dist_H(u,w) + dist_H(w,v) \leq d/2 + d/2 = d$.

   Therefore at least one of the induced subgraphs $H[B_v^{(d/2)}]$ and $H[B_u^{(d/2)}]$ has to contain less than $m/2$ edges. W.l.o.g. assume that this holds for $H[B_v^{(d/2)}]$. We now want to show that this cannot hold, because the balls $B_v^{(i)}$ grow exponentially as i increases. To prove this, we will use that H is a ϕ-expander.

   Let now m_i denote the number of edges in the induced subgraph $H[B_v^{(i)}]$. We have $m_1 \geq 1$ and $m_{d/2} \leq m/2$. Moreover,

   $$m_{i+1} = |E(B_v^{(i+1)})| \geq |E(B_v^{(i)})| + |E(B_v^{(i)}, B_v^{(i+1)})|.$$

   By the definition of $B_v^{(i)}$ we get that $E(B_v^{(i)}, B_v^{(i+1)}) = E(B_v^{(i)}, V(H) \setminus B_v^{(i)})$. Moreover, since H is a ϕ-expander,

   $$|E(B_v^{(i)}, V(H) \setminus B_v^{(i)})| \geq \phi \cdot \min\{deg_H(B_v^{(i)}), deg_H(V(H) \setminus B_v^{(i)})\}.$$

   We have $m_i \leq deg_H(B_v^{(i)}) \leq m_{i+1} < m/2$ and thus $\min\{deg_H(B_v^{(i)}), deg_H(V(H) \setminus B_v^{(i)})\} = deg_H(B_v^{(i)})$.

   All together we get $m_{i+1} \geq (1+\phi)m_i$. Thus,

   $$m_{2\phi^{-1}\log m+1} \geq (1+\phi)^{2\phi^{-1}\log m} \geq e^{\phi \cdot 2\phi^{-1}\log m/2} = m > m/2,$$

   where we use that $1 + \phi \geq e^{\frac{\phi}{1+\phi}} \geq e^{\frac{\phi}{2}}$. In particular, we get a contradiction to $m/2 > m_{d/2} \geq m_{2\phi^{-1}\log m+1}$.

2. Recall that with high probability for each G_i the weighted graph after the down-sampling preserves the cuts of G_i up to a factor of $(1 \pm \epsilon)$. In particular, since each edge in the down-sampling process receives the same weight w we get that for all $S \subseteq V(G_i)$

   $$(1-\epsilon)|E_{G_i'}(S, V(G_i)| \setminus S)| \leq \frac{1}{w}|E_{G_i}(S, V(G_i) \setminus S)| \leq (1+\epsilon)|E_{G_i'}(S, V(G_i) \setminus S)|.$$

   Since for any $v \in V(G_i)$ we have that $deg_{G_i}(v) = |E(\{v\}.V(G_i) \setminus \{v\})|$, we in particular get that for each $v \in G_i$,

   $$deg_{G_i'}(v) \leq \frac{1}{(1-\epsilon)w} deg_{G_i}(v).$$

   Now, let $\phi' = \frac{1-\epsilon}{1+\epsilon}\phi = \tilde{\mathcal{O}}(1)$. We claim that G_i' is a ϕ'-expander. Indeed, by the above considerations, and the fact that G_i is a ϕ-expander, for any $S \subseteq V(G_i)$ we get

   $$\begin{aligned}\phi' \min\{deg_{G_i'}(S), deg_{G_i'}(V(G_i) \setminus S)\} &\leq \frac{\phi}{(1+\epsilon)w} \min\{deg_{G_i}(S), deg_{G_i}(V(G_i) \setminus S)\} \\ &\leq \frac{1}{(1+\epsilon)w}|E_{G_i}(S, V(G_i) \setminus S)| \\ &\leq |E_{G_i'}(S, V(G_i) \setminus S)|.\end{aligned}$$

3. We first notice that if uv is an edge of G, then $dist_H(u,v) = \tilde{\mathcal{O}}(1)$. Indeed, let G_i be such that $uv \in E(G_i)$. Such a G_i exists, because by construction the G_i's partition the edges of G. By part 2, we get that G_i' is a $\tilde{\Omega}(1)$-expander. Therefore, since $u, v \in V(G_i')$ by part 1 and the fact that H is a graph obtained by adding edges to G_i', we get that $dist_H(u,v) \leq dist_{G_i'}(u,v) = \tilde{\mathcal{O}}(1)$.

   Now, let $u, v \in V(G)$ be arbitrary and let $u = w_0, w_1, \ldots, w_d = v$ be a shortest path between and v. Then, by the triangle inequality and the fact that each $w_i w_{i+1}$ is an edge in G
$$dist_H(u,v) \leq \sum_{i=0}^{d-1} dist_H(w_i, w_{i+1}) \leq \sum_{i=0}^{d-1} \tilde{\mathcal{O}}(1) = d \cdot \tilde{\mathcal{O}}(1).$$

   So H is a $\tilde{\mathcal{O}}(1)$-spanner.

3 Cut-Preserving Sparsifiers for Graphs with Large Min-Cut

Similarly as for the K_n and expanders in the lecture, we want to simply randomly sample some edges from G. Specifically, we set $p = \frac{10 \cdot \log n}{\epsilon^2 k}$ and return a graph H obtained from G by making every edge of G appear in H independently, with probability p. Moreover, we set the weight of each edge in H to p^{-1}.

Let us now show that the graph H obtained in this way is indeed a cut-preserving sparsifier. Let first X denote the number of edges in H. Then, $X \sim Bin(m,p)$, $\mathbb{E}[X] = mp$ and thus by Chernoff
$$\Pr[X > 20 \cdot \frac{\log n}{\epsilon^2 m} m] \leq \Pr[X - \mathbb{E}[X] \geq \mathbb{E}[X]] \leq e^{\mathbb{E}[X]} = o(1).$$

Now let us consider an arbitrary cut $(S, V(G) \setminus S)$ in G and let $\alpha \geq 1$ be such that this cut has size αk. Let Y denote the number of edges between S and $V(G) \setminus S$ in H such that $|E_H(S, V(H) \setminus S)| = p^{-1} Y$. We have $Y \sim Bin(\alpha k, p)$ and $\mathbb{E}[Y] = \alpha k p$. Then, again by Chernoff. we get that
$$\begin{aligned}
\Pr[||E_H(S, V(H) \setminus S)| - \alpha k| > \epsilon \alpha k] &= \Pr[|Y - \mathbb{E}[Y]| > \epsilon \mathbb{E}[Y]] \\
&\leq 2\exp(-\frac{\epsilon^2 \mathbb{E}[Y]}{3}) \\
&\leq 2\exp(-\frac{10\alpha}{3} \log n) \\
&\leq n^{-3\alpha},
\end{aligned}$$

where the last inequality holds for n large enough.

We now want to use union bound to bound the probability that some cut C is not preserved in H. We get
$$\begin{aligned}
\Pr[\exists C \text{ not preserved in } H] &\leq \sum_{C \text{ cut in } G} \Pr[C \text{ not preserved in } H] \\
&\leq \sum_{s=k}^{n^2} (2n)^{\frac{s}{k}} \cdot n^{-3\frac{s}{k}} \\
&\leq n^2 \cdot n^{-2.5} = o(1),
\end{aligned}$$

where we used the probability calculated above and the bound $(2n)^{\frac{s}{k}}$ on the number of cuts of size $s = \frac{s}{k} \cdot k$ in G. In particular, with high probability all the cuts are preserved in H and H has at most $20 \cdot \frac{\log n}{\epsilon^2 k} m$ edges.