6.5240 Sublinear Time Algorithms October 22nd, 2026

Lecture 12: LCA for Graph Spanners

Lecturer: Ronitt Rubinfeld *Scribe: Sitta Tantikul*

Topics covered.

- Greedy construction of sparse $(2k-1)$-spanners and the girth bound.
- Local computation algorithms (LCAs) for 3-spanners.
- Bucketing and degree classes to obtain $\tilde{O}(n^{3/2})$ edges with $\tilde{O}(\sqrt{n})$ probes per query.

1 Graph Spanners

Consider an undirected, unweighted graph $G = (V, E)$ with $|V| = n$.

Definition 1.1 (t-spanner) *A* t-spanner *of G is a subgraph $H = (V, E')$ on the same set of vertices, with $E' \subseteq E$, such that*

$$\text{dist}_H(u, v) \leq t \cdot \text{dist}_G(u, v) \qquad \forall u, v \in V.$$

1.1 Greedy centralized construction

Fix an arbitrary order on the edges of G. Starting from $H = (V, \emptyset)$, process the edges $\{u, v\} \in E$ one at a time and add $\{u, v\}$ to H iff

$$\text{dist}_H(u, v) \geq 2k.$$

($\text{dist}_H(u, v) = \infty$ if u and v are disconnected.)

Proposition 1.2 ([ADD+93]) *The resulting graph H is a $(2k-1)$-spanner with*

$$|E(H)| = O\left(n^{1+\frac{1}{k}}\right).$$

Proof First consider an edge $\{u, v\} \in E$ that is not added to H. At the moment it is considered, there is already a path from u to v in H of length at most $2k-1$. Since edges are only added later, this path remains in the final graph. Replacing every edge on a shortest path in G by such a path in H gives

$$\text{dist}_H(x, y) \leq (2k-1)\text{dist}_G(x, y)$$

for all $x, y \in V$.

For the size bound, H has no cycle of length at most $2k$. Indeed, if such a cycle existed, consider the last edge $\{u, v\}$ of the cycle that was added. The other edges of the cycle were already present, so they gave a path from u to v of length at most $2k-1$, contradicting the rule for adding $\{u, v\}$. Hence

$$\text{girth}(H) \geq 2k + 1.$$

Let $\bar{d} = 2|E(H)|/n$ be the average degree of H. The Moore bound for irregular graphs [AHL02] states that if a graph has girth at least $2k+1$, then

$$n \geq 1 + \bar{d}\sum_{i=0}^{k-1}(\bar{d}-1)^i.$$

In particular, for $\bar{d} \geq 2$,

$$n \geq \bar{d}(\bar{d}-1)^{k-1} \geq (\bar{d}-1)^k \quad \Rightarrow \quad \bar{d} \leq n^{1/k} + 1.$$

Therefore

$$|E(H)| = \frac{n\bar{d}}{2} \le \frac{1}{2}n^{1+1/k} + \frac{1}{2}n = O\Big(n^{1+1/k}\Big).$$

(The case $\bar{d} < 2$ is immediate.) Thus the greedy construction contains $O(n^{1+1/k})$ edges. ■

Conjecture 1.3 (Erdős Girth Conjecture) *For every positive integer k, there exist arbitrarily large n-vertex graphs with girth at least $2k+1$ and*

$$\Omega\Big(n^{1+\frac{1}{k}}\Big)$$

edges.

2 Local Computation for a 3-Spanner

Taking $k = 2$ in 1.2, every graph has a 3-spanner with $O(n^{\frac{3}{2}})$ edges. Our goal today is a local computation algorithm (LCA) which answers whether a queried edge belongs to one fixed sparse 3-spanner, using only $\tilde{O}(\sqrt{n})$ probes per query [PRVY19].

Probe model. Consider each vertex with a unique ID in $[n]$. We write $N(v)[i]$ for the i-th entry of the incidence list of v. We allow the following probes:

- **Degree probe:** given v, return $\deg(v)$.
- **Neighbor probe:** given v and i, return $N(v)[i]$ (or $\perp$ if $i > \deg(v)$).
- **Adjacency probe:** given $\{u, v\}$, determine whether $\{u, v\} \in E$; if so, return the positions of u in $N(v)$ and of v in $N(u)$.

Remark. [PRVY19] Also showed an $\Omega(\sqrt{n})$ probe lower bound for this problem, so their $\tilde{O}(\sqrt{n})$-probe algorithm is optimal up to polylogarithmic factors.

3 Attempt I: One Global Set of Centers

We first try separate vertices by degree into:

$$\text{light: } \deg(v) \le \sqrt{n}, \qquad \text{heavy: } \deg(v) > \sqrt{n}.$$

Keep every edge incident to a light vertex. This contributes at most $n\sqrt{n} = n^{\frac{3}{2}}$ edges.

Next, we independently sample each vertex as a *center* with probability $p = \Theta\Big(\frac{\log n}{\sqrt{n}}\Big)$, and let S be the set of centers. Thus $|S| = \Theta(\sqrt{n}\log n)$ with high probability.

A center $s \in S$ defines the cluster consisting of the vertices adjacent to s.

Observation 3.1 *With high probability, every heavy vertex is adjacent to at least one center.*

Proof Fix a heavy vertex u. Since $\deg(u) > \sqrt{n}$,

$$\Pr[N(u) \cap S = \emptyset] \le (1-p)^{\sqrt{n}} \le e^{-p\sqrt{n}} = n^{-\Theta(1)}.$$

Choosing the constant in p sufficiently large and taking a union bound over all heavy vertices u proves the claim. ■

Construction. For every vertex v and every center s whose cluster is adjacent to v, keep exactly one edge from v into the cluster of s. Fix random ranks on the edges once and for all, and use the minimum-rank eligible edge to break ties. Also keep every edge between a heavy vertex and its adjacent centers.

Correctness. Suppose $\{u, v\}$ is omitted. Both endpoints are heavy. Pick any center $s \in S \cap N(u)$. Since v has a neighbor in the cluster of s (namely u), it keeps some edge $\{v, w\}$ into that cluster. Then

$$v - w - s - u$$

is a path of length at most 3 in the spanner.

Size. For every v, there is at most one kept edge from v into each cluster, so

$$|E(H)| \leq n \cdot |S| = \tilde{O}(n^{3/2}).$$

Why the local query is too slow. Given a query $\{u, v\}$, suppose we want to determine whether it is the chosen edge from v into one of the clusters containing u.

- Scan the centers for adjacency to u: $\tilde{O}(\sqrt{n})$ probes.
- Scan v's incidence list to check whether an earlier edge (under the fixed edge ranks) already connects v to the same cluster: up to $\deg(v)$ probes.

Thus the query complexity is

$$\tilde{O}(\sqrt{n} + \deg(v)),$$

which can be as large as $\tilde{O}(n)$ and might not be sublinear.

4 Fix Attempt via Bucketing

Following the $\tilde{O}(\sqrt{n})$ LCA construction of Arviv, Chung, Levi, and Pyne [ACLP21], we first divide each incidence list into consecutive buckets of size $\sqrt{n}$.

Probe complexity. To answer a query $\{u, v\}$:

- find the centers whose clusters contain u by scanning S;
- find the bucket of $N(v)$ containing u;
- scan only that bucket to determine whether $\{u, v\}$ is the minimum-rank edge in the bucket into one of those clusters.

Hence

$$\#\text{probes} = \tilde{O}(\sqrt{n}) + \tilde{O}(\sqrt{n}) = \tilde{O}(\sqrt{n}).$$

But the size is now too large. A vertex may have as many as $\sqrt{n}$ buckets, and there are $\Theta(\sqrt{n} \log n)$ clusters. The number of edges becomes

$$n \cdot \sqrt{n} \cdot \sqrt{n} \log n = O(n^2 \log n).$$

Hence, we lose the guarantee we had on the $\tilde{O}(n^{3/2})$ size of H.

5 Refined Fix via Degree Classes

The key additional idea of [ACLP21] is to sample fewer centers for higher degree classes to accommodate the high number of buckets.

5.1 Degree classes and center sets

For $i = 0, 1, \ldots, \lfloor \log_2 \sqrt{n} \rfloor$ define

$$V_i = \left\{ v : \deg(v) \in (2^i \sqrt{n}, 2^{i+1} \sqrt{n}] \right\}.$$

(Vertices of degree $\leq \sqrt{n}$ are light and are handled separately.)

For each class V_i, sample a separate set of centers S_i of expected size $\mathbb{E}[|S_i|] = \Theta\left(2^{-i} \sqrt{n} \log n\right)$. Equivalently, each vertex may be placed in S_i independently with the corresponding probability.

For $v \in V_i$, write $C_v = S_i \cap N(v)$ for the centers of v.

Claim 5.1 *With high probability,* $1 \leq |C_v| \leq O(\log n)$ *for every heavy vertex* $v \in V_i$.

Proof For $v \in V_i$, the expected number of adjacent centers is

$$\mathbb{E}[|C_v|] = \Theta\left(2^{-i} \sqrt{n} \log n \cdot \frac{\deg(v)}{n}\right) = \Theta(\log n).$$

A Chernoff bound gives both the lower and upper tails, and union bounding over all $v \in V_i$ gives the claim. ■

Thus every heavy vertex belongs to at most $O(\log n)$ clusters of its own degree class.

5.2 Final construction

Fix a total order on vertices: $u \succ v \iff \deg(u) > \deg(v) \vee [\deg(u) = \deg(v) \text{ and } ID(u) > ID(v)]$ with fixed random center sets and tie-breaking orders are reused across every query.

Add the following edges to H:

1. If v is light, keep every edge incident to v.
2. For every heavy u, keep the edge $\{u, s\}$ for every $s \in C_u$.
3. Fix heavy vertex v, divide neighbors $N(v)$ into buckets of $\sqrt{n}$ consecutive entries. In each bucket, and for every center s belonging to a class at least as large as the class of v, keep the first edge $\{v, u\}$ in that bucket such that
$$u \succ v \qquad \text{and} \qquad s \in C_u.$$

Equivalently: each vertex keeps one edge per bucket into each adjacent cluster of equal or higher degree class.

5.3 Local query algorithm

Given a query $\{u, v\} \in E$, relabel the endpoints so that $u \succ v$.

1. If v is light, return YES.
2. Let $u \in V_j$ and $v \in V_i$. If $v \in C_u$ or $u \in C_v$, return YES.

3. Compute $C_u = S_j \cap N(u)$ by scanning the candidate centers in S_j. If $C_u = \emptyset$, return YES as a safe failure-event fallback.

4. Use the adjacency probe to find the position of u in $N(v)$ and hence the bucket containing it.

5. For every $s \in C_u$, scan only the entries preceding u in this bucket. If none of them is a vertex $w \succ v$ with $s \in C_w$, then return YES.

6. Otherwise return NO.

5.4 Probe complexity

Claim 5.2 *With high probability, every query uses $\tilde{O}(\sqrt{n})$ probes.*

Proof Suppose $u \in V_j$. Scanning the candidate centers in S_j costs at most

$$|S_j| = O(2^{-j}\sqrt{n}\log n) = O(\sqrt{n}\log n).$$

By Claim 5.1, $|C_u| = O(\log n)$ with high probability. For each $s \in C_u$, we inspect at most one bucket of size $\sqrt{n}$. Thus the bucket work costs $O(\sqrt{n}\log n)$ probes, and therefore the total is

$$O(\sqrt{n}\log n) = \tilde{O}(\sqrt{n}).$$

■

5.5 Size

Claim 5.3 *With high probability $|E(H)| = \tilde{O}(n^{3/2})$.*

Proof Edges incident to light vertices contribute at most $n\sqrt{n}$ edges, and center edges contribute only $O(n\log n)$ edges with high probability.

Now fix $v \in V_i$. Since $\deg(v) \le 2^{i+1}\sqrt{n}$ the incidence list of v has at most 2^{i+1} buckets.

A neighbor $u \succ v$ belongs to some class V_j with $j \ge i$ (up to the tie-breaking order inside one class), and its centers lie in S_j. Therefore the total number of centers whose clusters can be relevant to v is at most

$$\sum_{j \ge i} |S_j| = \tilde{O}(2^{-i}\sqrt{n}).$$

Since v keeps at most one edge per (bucket,eligible center) pair, the number of Rule (3) edges charged to v is at most

$$2^{i+1} \cdot \tilde{O}(2^{-i}\sqrt{n}) = \tilde{O}(\sqrt{n}).$$

Summing over all n vertices gives

$$|E(H)| = \tilde{O}(n^{3/2}).$$

■

5.6 Stretch

Claim 5.4 *The resulting graph is a 3-spanner.*

Proof Let $\{u, v\} \in E$ be omitted, with $u \succ v$. Then v is heavy and $C_u \neq \emptyset$. Fix $s \in C_u$.

Since $\{u, v\}$ was not the first eligible edge in its bucket into the cluster of s, there is an earlier vertex w in the same bucket with

$$w \succ v \qquad \text{and} \qquad s \in C_w.$$

Choose the first such w. By Rule (3), $\{v, w\} \in H$. By Rule (2), $\{w, s\}, \{s, u\} \in H$. Hence

$$v - w - s - u$$

is a path of length 3, so

$$\mathrm{dist}_H(u, v) \leq 3.$$

■

Summary. The three stages illustrate the two competing requirements:

	Probe complexity	**Number of edges**
One global center set	$\tilde{O}(\sqrt{n} + \deg(v))$ — too large	$\tilde{O}(n^{3/2})$ — good
Add $\sqrt{n}$-sized buckets	$\tilde{O}(\sqrt{n})$ — good	$O(n^2 \log n)$ — too large
Add degree classes and shrinking center sets	$\tilde{O}(\sqrt{n})$ — good	$\tilde{O}(n^{3/2})$ — good

Thus we obtain an LCA for a 3-spanner with $\tilde{O}(n^{3/2})$ edges using $\tilde{O}(\sqrt{n})$ probes per query.

References

[ACLP21] Rubi Arviv, Lily Chung, Reut Levi, and Edward Pyne. Improved local computation algorithms for constructing spanners. arXiv:2105.04847, 2021.

[ADD+93] Ingo Althöfer, Gautam Das, David Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9:81–100, 1993.

[AHL02] Noga Alon, Shlomo Hoory, and Nathan Linial. The moore bound for irregular graphs. *Graphs and Combinatorics*, 18(1):53–57, 2002.

[PRVY19] Merav Parter, Ronitt Rubinfeld, Ali Vakilian, and Anak Yodpinyanee. Local computation algorithms for spanners. In *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*, volume 124 of *LIPIcs*, pages 58:1–58:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019.