Local Computation Algorithms for Spanners

Merav Parter, Ronitt Rubinfeld, Ali Vakilian and Anak Yodipinyanee
Graph Spanners

A subgraph $H \subseteq G$ is a $k$ spanner if: $\text{dist}_H(u,v) \leq k \cdot \text{dist}_G(u,v), \forall u,v$

**Fact:** Every $n$-vertex graph has $(2k - 1)$ spanner with $O(n^{1+1/k})$ edges.

- Numerous Applications:
  Routing, synchronizers, SDD, spectral sparsifiers ...

- Various of Computational Settings:
  Distributed, parallel, dynamic, streaming ...
LCA for Spanners

**The setting:** Huge graph that cannot be stored on main memory

**Goal:** Implement fast (local) access to sparse spanner

LCA decides **locally** if a given edge $e$ is in the spanner without ever computing the entire spanner.

Making primitive probes (neighbors)

*Consistent* with respect to a unique spanner $H \subseteq G$
The Model [Alon, Rubinfeld, Vardi, Tamir’12]

Input graph: $N(u_1), N(u_2), ..., N(u_n)$

Neighbor probe: “what’s the $i$’th neighbor of $u$?”

"$w$" 

“$v$ is the $j$’th neighbor of $u$”

Adjacency probe: Are $u$ and $v$ neighbors?

LCA

- random string
- work space

Unique output (spanner)

- no preprocessing
- no auxiliary info

Slide modified from Rubinfeld
Swarms of LCA

Input Graph

Initially share random string

Output y

Afterwards compute independently

Complexity measure: number of probes (here also quality of spanner)

Slide from Rubinfeld
Previous Work

LCA for sparse subgraphs [Levi-Ron-Rubinfeld 14’]: Provide fast random access to a sparse connected subgraph

Query: does $e$ belong to a sparse subgraph $H \subseteq G$?

- If $H \subseteq G$ is a tree, probe complexity of $\Omega(n)$
- Relax to $|H| = (1 + \epsilon)n$ edges
- LB of $\Omega(\sqrt{n})$ probes for bounded-deg graphs
- UB for special graph families: subexponential-growth, excluded-minor.
<table>
<thead>
<tr>
<th>Year</th>
<th>Graph Family</th>
<th># Edges</th>
<th>Stretch</th>
<th>Probe s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Levi-Ron-Rubinfeld’14</td>
<td>▪ Expanders</td>
<td>$(1 + \epsilon)n$</td>
<td>$\sim \sim$</td>
<td>$O(\sqrt{n})$</td>
</tr>
<tr>
<td></td>
<td>▪ Subexponential growth</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Levi-Ron’15</td>
<td>Minor-free</td>
<td>$(1 + \epsilon)n$</td>
<td>Poly($\Delta$, $1/\epsilon$)</td>
<td>Poly($\Delta$, $1/\epsilon$)</td>
</tr>
<tr>
<td>Levi-Ron-Rubinfeld’16</td>
<td>Minor-free</td>
<td>$(1 + \epsilon)n$</td>
<td>$O(\log \Delta/\epsilon)$</td>
<td>Poly($\Delta$, $1/\epsilon$)</td>
</tr>
<tr>
<td>Levi-Moshkovitz-Ron-Rubinfeld-Shapira’17</td>
<td>Bounded Expansion</td>
<td>$(1 + \epsilon)n$</td>
<td>$\exp(\frac{1}{\epsilon})$</td>
<td>$\exp(\frac{1}{\epsilon})$</td>
</tr>
<tr>
<td>Lenzen-Levi’18</td>
<td>General graphs (max-deg $\Delta$)</td>
<td>$(1 + \epsilon)n$</td>
<td>$\tilde{O}(\Delta/\epsilon)$</td>
<td>$\tilde{O}(\Delta^4 \cdot n^{2/3})$</td>
</tr>
<tr>
<td>New</td>
<td>General graphs (max-deg $\Delta$)</td>
<td>$\tilde{O}(n^{1+1/k})$</td>
<td>$O(k^2)$</td>
<td>$\tilde{O}(\Delta^4 \cdot n^{2/3})$</td>
</tr>
<tr>
<td>New</td>
<td>General graphs</td>
<td>$\tilde{O}(n^{1+1/r})$</td>
<td>$r \in {2,3}$</td>
<td>$\tilde{O}(n^{1+1/2r})$</td>
</tr>
</tbody>
</table>
**Our Results**

*Spanner LCA* implements oracle access to a unique sparse spanner

<table>
<thead>
<tr>
<th>Stretch</th>
<th># Edges</th>
<th># Probes</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-spanner</td>
<td>$\tilde{O}(n^{3/2})$</td>
<td>$\tilde{O}(n^{3/4})$</td>
</tr>
<tr>
<td>5-spanner</td>
<td>$\tilde{O}(n^{4/3})$</td>
<td>$\tilde{O}(n^{5/6})$</td>
</tr>
<tr>
<td>$O(k^2)$-spanner</td>
<td>$\tilde{O}(n^{1+1/k})$</td>
<td>$\tilde{O}(\Delta^4 n^{2/3})$</td>
</tr>
<tr>
<td>$O(n)$ spanner</td>
<td>$o(m)$</td>
<td>$\Omega(\min(\sqrt{n}, n^2/m))$</td>
</tr>
</tbody>
</table>
First Attempt: LCA through Distributed Algorithm

Fact: Distributed constructions of \((2k-1)\) spanners in \(k\) rounds.

[Baswana-Sen’06, Elkin-Neiman ’17]

- Translates into a LCA with probe complexity of \(\Omega(\Delta^k)\)
- **Superlinear** for the interesting case when \(\Delta = \Omega(n^{1/k})\)
- **Goal**: *sublinear* (in \(n\)) probe complexity, not even clear how to achieve \(O(\Delta)\)
Clustering Alg. for 3-Spanner (Baswana-Sen)

Desired Output: 3-spanner $H \subseteq G$ with $\tilde{O} \left( \frac{n^3}{2} \right)$ edges.

Def: A vertex $v$ is high-deg if $\deg(v) \geq \sqrt{n}$

- Add all edges of low-deg vertices to $H$
- Sample $S \subseteq V$ of $O(n^{1/2} \log n)$ centers
- Connect high-deg $v$ to $s(v) \in S \cap N(v)$
- Every high-deg $v$ adds one edge to each neighboring stars.
Clustering Alg. for 3-Spanner (Baswana-Sen)

The algorithm has several degrees of freedom:

- Picking the center of high-deg vertex
- Picking the edge to add to each neighboring star

The LCA will fix these selections to obtain small probe complexity
First Approach for Spanner LCA

- Picking the **first** sampled neighbor as a center (for high-deg nodes)
- Picking the **first** edge to each neighboring star

\[ N(v) \quad v_1, u_2, \ldots v_j, \ldots, v_\ell = u, \ldots, \]
\[ s(v) \]
\[ N(u) \quad u_1, u_2, \ldots u_i, \ldots, u_k = v, \ldots, \]
\[ s(u) \]

By storing poly-log n random bits know if \( s \in S \)
The First Approach

**Query:** edge \((u, v)\)

**Answer:** Yes iff \((u, v) \in H\)

- If \(\min(\text{deg}(u), \text{deg}(v)) \leq \sqrt{n}\): “yes”

Remains to handle edges between high-deg vertices

- In \(\text{deg}(v)\) probes can compute the center \(s(v)\)
- Should “yes” if \(v\) is the first neighbor of \(u\) in the cluster of \(s(v)\)
The First Approach

Query: edge \((u, v)\), interesting case: \(\deg(u), \deg(v) > \sqrt{n}\)

For each \(u's\) neighbor \(u_i\) appearing before \(v\) in \(N(u)\) check if \(s(v)\) is the center of \(u_i\)

Probe complexity \(O(\Delta^2)\)
The First Multiple Center Approach

The centers of $u$ are all sampled vertices in the first block of $\sqrt{n}$ neighbors.

- By Chernoff, w.h.p., high-deg vertex has $\Theta(\log n)$ centers.

- Testing membership to a cluster: “does $w$ belong to the cluster of $s$”? Can be done with a single adj. probe!

$w_1, w_2, ..., w_j, ..., w_\ell, ...$

\[ \sqrt{n} \]

$w \in C(s)\ ?$

Adj. $\langle w, s \rangle$

Yes, if $j \leq \sqrt{n}$
LCA with $\deg(u)$ Probes

Query: edge $(u, v), \deg(u), \deg(v) > \sqrt{n}$

Compute $S(v)$: the first centers of $v$

$v_1, v_2, \ldots, v_j, \ldots, v_\ell, \ldots$

$N(u)$

$u_1, u_2, \ldots, u_i, \ldots, u_k = v, u_{k+1}, \ldots$

If $u \in S(v)$, say "yes".

For every $s \in S(v)$ and $u_i < v$:

Check if $u_i \in C(s)$ (single adj. query)

If $\exists s' \in S(v)$ for which no $u_i < v$ is in $C(s')$, say “yes”

Probe complexity: $|S(v)| \cdot \deg(u) = O(\deg(u) \cdot \log n)$
LCA with $\text{deg}(u)$ Probes

Size analysis:
An edge $(u, u_i) \in H$ only if $S(u_i)$ has a new center not in $\bigcup_{j<i} S(u_j)$.

Stretch analysis:
An edge $(u, u_i) \notin H$, with $s' \in S(u_i)$ $u_j$ be the first in $N(u)$ s.t $s' \in S(u_j)$

$(u, u_j) \in H$

$\text{dist}_H(u, u_i) \leq 3$
3-Spanner with Sublinear Probe Complexity

- Edge query \((u, v)\) with \(\min(\deg(u), \deg(v)) \leq n^{3/4}\)

New approach for very-high deg vertices with \(\deg(v) \geq n^{3/4}\)

Decompose \(N(u)\) into “independent” blocks of size \(n^{3/4}\)

\[ N(u) \]

\[ u_i, \ldots, v, \ldots u_j \]

\[ n^{3/4} \]

Given query \((u, v)\) decide based on the block of \(v\) in \(N(u)\)
3-Spanner with Sublinear Probe Complexity

- Sample $S \subseteq V$ of $O(n^{1/4} \log n)$ centers
- Define first-centers $S(u) = S \cap N_1(u)$

**Query** $(u, v)$:
- compute $N_i(u)$ and first-centers $S(v)$
- For every $u_j < v$ and $s \in S(v)$:
  - Check if $u_j \in C(s)$ (single adj. query)
  - If $\exists s' \in S(v)$ for which no $u_j < v$ is in $C(s')$, say “yes”
Analysis

Query \((u, v)\):

- Compute \(N_i(u)\) and first-centers \(S(v)\)
- For every \(u_j < v\) and \(s \in S(v)\):
  - Check if \(u_j \in C(s)\) (single adj. query)
- If \(\exists s' \in S(v)\) for which no \(u_j < v\) is in \(C(s')\), say “yes”

Probe complexity: \(|S(v)| \cdot |N_i(u)| = O(n^{3/4} \log n)\)
Query \((u, v)\):

- compute \(N_i(u)\) and first-centers \(S(v)\)
- For every \(u_j < v\) and \(s \in S(v)\):
  - Check if \(u_j \in C(s)\) (single adj. query)
  - If \(\exists s' \in S(v)\) for which no \(u_j < v\) is in \(C(s')\), say “yes”

**Size:** In each block \(N_i(u)\), add one edge per sampled center.
Overall, \(|H| = O(n \cdot n^{1/4} \cdot n^{1/4} \cdot \log n)\)
Spanner LCA with Stretch $O(k^2)$

- Sample a set $S \subseteq V$ of $O(n^{2/3})$ centers.
- Sparse vertices $N_k(v) \cap S = \emptyset$, hence w.h.p. $|N_k(v)| = O(n^{1/3})$
- Dense vertices $N_k(v) \cap S \neq \emptyset$

Spanner $H_S$ for the sparse subgraph $(V_S \times V) \cap E$

- Collect the $k$-neighborhood of sparse vertex $v$
- Apply a $k$-round distributed algorithm on $N_k(v)$
- Probe complexity $O(\Delta n^{2/3})$
Spanner LCA for the Dense Subgraph

- Sample $S$ of $O(n^{2/3})$ centers, Dense vertices $N_k(v) \cap S \neq \emptyset$
- Assign nodes in $S$ random IDs in $[1, n]$

Voroni-cells centered at $S$

- Add a BFS tree (depth $k$) in each cell
- All dense vertices are covered

How to connect adjacent cells?
Spanner LCA for the Dense Subgraph

- Subdivide each Voronoi cell into clusters with \( O(n^{1/3}) \) vertices
- \( \tilde{O}(n^{2/3}) \) clusters
- Sample \( \tilde{O}(n^{1/3}) \) centers \( S' \subseteq S \)

All clusters of a sampled Voronoi cell are marked.
Spanner LCA for the Dense Subgraph

- $\tilde{O}(n^{1/3})$ marked clusters.
- Connect each cluster to all marked neighboring clusters ($\tilde{O}(n)$ edges)
- Connect each cluster with no marked neighbor, to all of its neighbors
Spanner LCA for the Dense Subgraph

For each marked cluster $C$ and cluster $C'$:

Connect $C'$ to the Voronoi cells of the $\widetilde{O}(n^{1/k})$ minimum center-IDs in $N_{Vor}(C) \cap N_{vor}(C')$.

Overall this adds $\widetilde{O}(n^{2/3} \cdot n^{1/3} \cdot n^{1/k})$ edges.
Analysis idea:

Inductive argument (based on random ranks) to show stretch in $H_{\text{vor}}$ is $O(k)$

Since each Vor cell has depth $k$ tree, yields a total stretch of $O(k^2)$
### Summing Up and Open Problems

<table>
<thead>
<tr>
<th>Stretch</th>
<th># Edges</th>
<th># probes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2r - 1, r \in {2,3}$</td>
<td>$\tilde{O}(n^{1+r})$</td>
<td>$\tilde{O}(n^{1-1/2r})$</td>
</tr>
</tbody>
</table>

Open for $r \geq 4$

| $O(k^2)$-spanner | $\tilde{O}(n^{1+1/k})$ | $\tilde{O}(\Delta^4n^{2/3})$ |

Improve dependency on $\Delta$

| $O(n)$ spanner | $o(m)$ | $\Omega(\min(\sqrt{n}, n^2/m))$ |

Toda! Stretch-sensitive lower bounds