Local Computation Algorithms for Spanners

Merav Parter, Ronitt Rubinfeld, Ali Vakilian and Anak Yodipinyanee

מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

Graph Spanners

A subgraph $\boldsymbol{H} \subseteq \boldsymbol{G}$ is a $\boldsymbol{k}$ spanner if: $\boldsymbol{dist_H(u,v) \leq k \cdot dist_G(u,v), \forall u,v}$

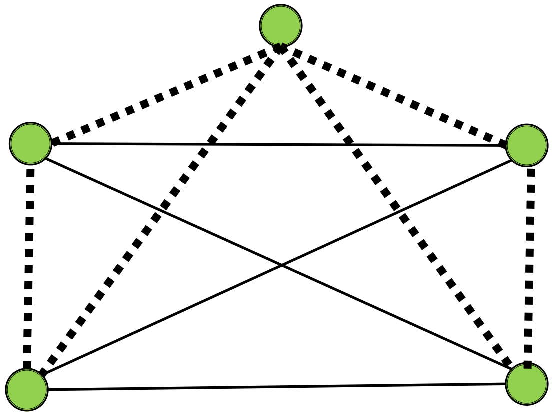
Fact: Every n-vertex graph has $(\boldsymbol{2k-1})$ spanner with $\boldsymbol{O(n^{1+1/k})}$ edges.

- Numerous Applications:

Routing, synchronizers, SDD, spectral sparisfiers …

- Various of Computational Settings:

Distributed, parallel, dynamic, streaming …

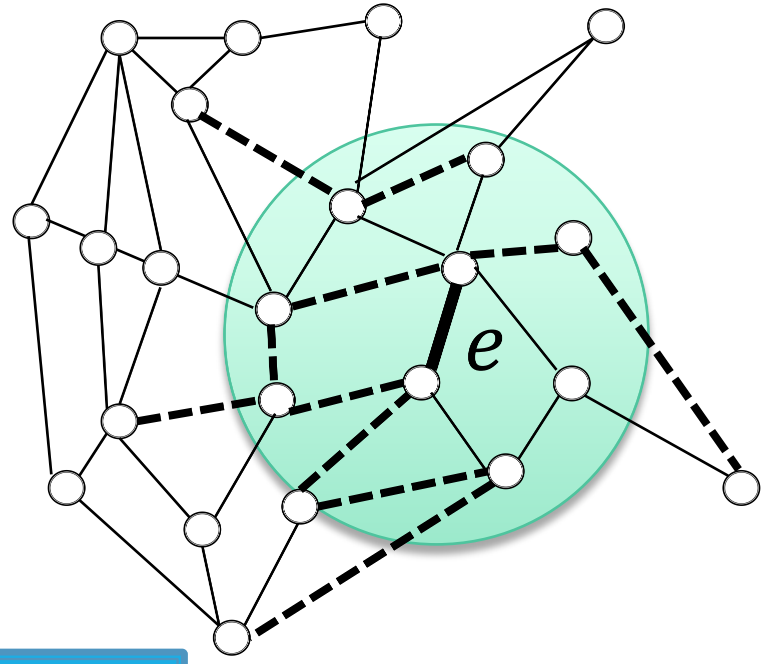


LCA for Spanners

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: $N(u_1), N(u_2), \ldots, N(u_n)$

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

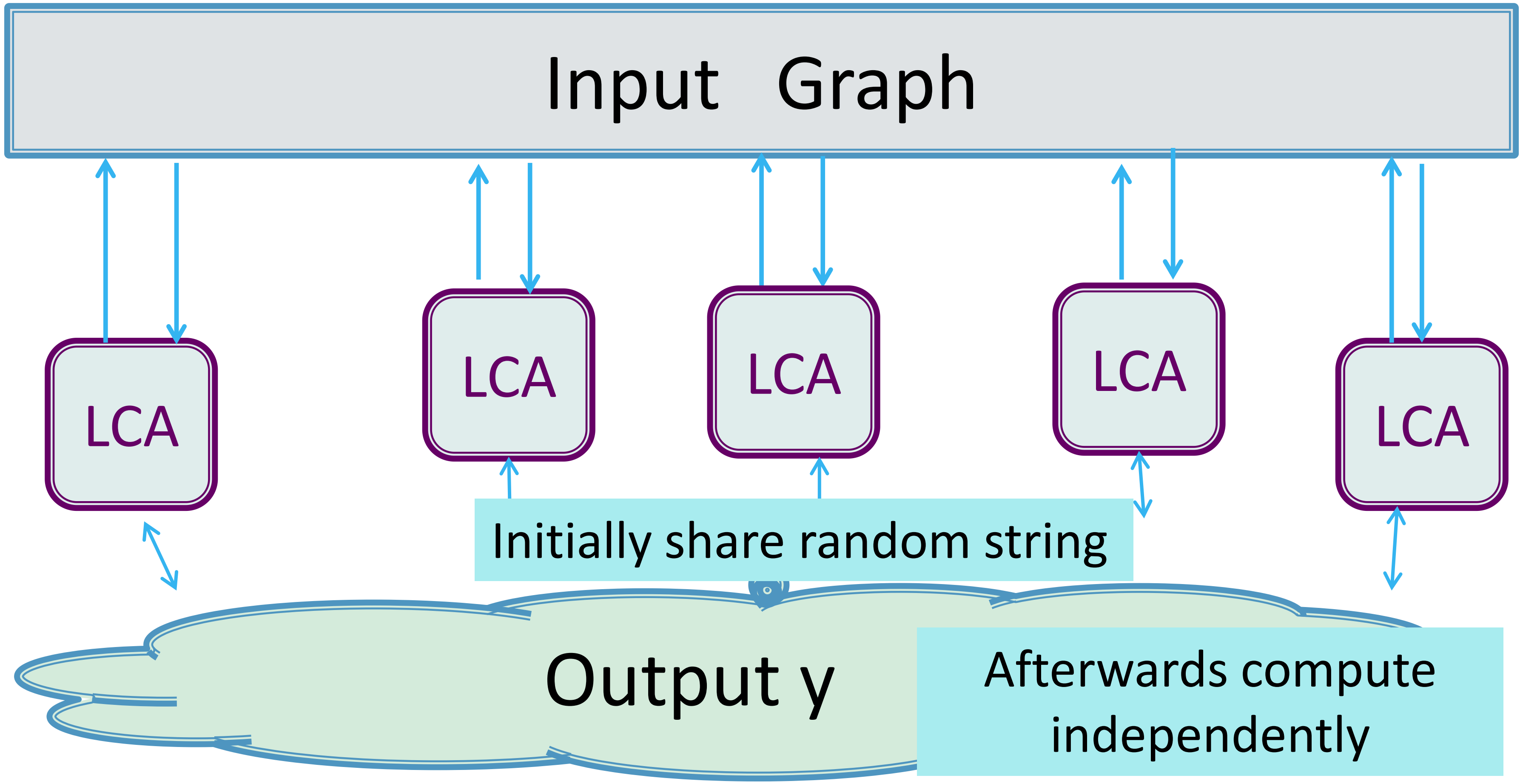
work space

Unique output (spanner)

- no preprocessing
- no auxiliary info

Slide modified from Rubinfeld

Swarms of LCA



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

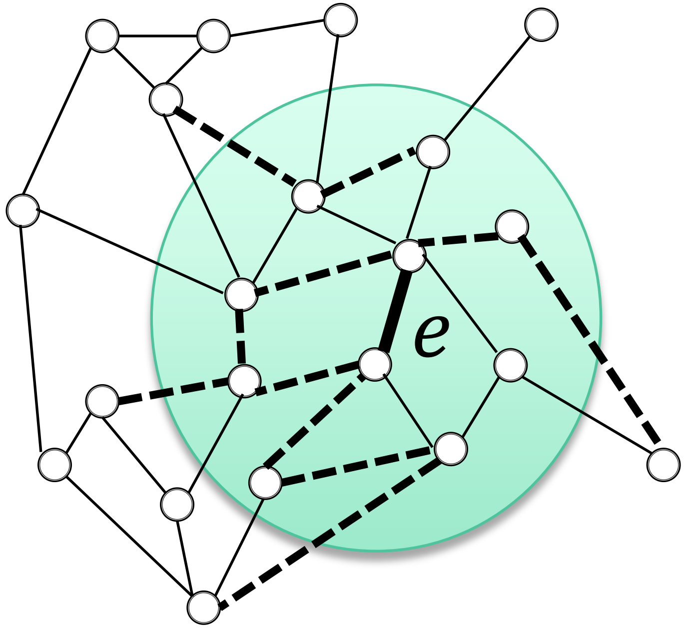
Slide from Rubinfeld

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.



	Graph Family	# Edges	Stretch	Probe s
Levi-Ron-Rubinfeld'14	▪ Expanders ▪ Subexponential growth	$(1+\epsilon)n$	--- -	$O(\sqrt{n})$
Levi-Ron'15	Minor-free	$(1+\epsilon)n$	$\mathrm{Poly}(\Delta, 1/\epsilon)$	$\mathrm{Poly}(\Delta, 1/\epsilon)$
Levi-Ron-Rubinfeld'16	Minor-free	$(1+\epsilon)n$	$\mathbf{O(\log \Delta/\epsilon)}$	$\mathrm{Poly}(\Delta, 1/\epsilon)$
Levi-Moshkovitz-Ron-Rubinfeld-Shapira'17	Bounded Expansion	$(1+\epsilon)n$	$\mathrm{Exp}(\frac{1}{\epsilon})$	$\mathrm{Exp}(\frac{1}{\epsilon})$
Lenzen-Levi'18	General graphs (max-deg Δ)	$(1+\epsilon)n$	$\tilde{O}(\Delta/\epsilon)$	$\tilde{O}(\Delta^4 \cdot n^{2/3})$
New	General graphs (max-deg Δ)	$\tilde{\boldsymbol{O}}(\boldsymbol{n^{1+1/k}})$	$\boldsymbol{O(k^2)}$	$\tilde{O}(\Delta^4 \cdot n^{2/3})$
New	General graphs	$\tilde{O}(n^{1+1/r})$	$r \in \{2,3\}$	$\tilde{O}(n^{1+1/2r})$

Our Results

Spanner LCA implements oracle access to a unique sparse spanner

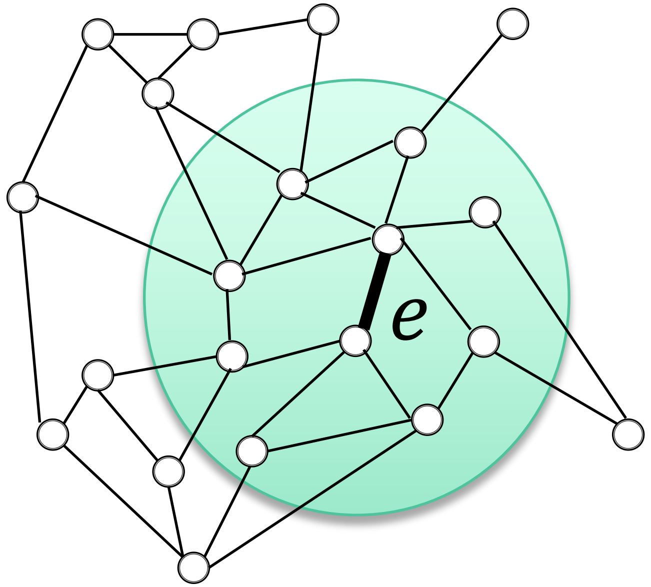
Stretch	# Edges	# Probes	
3-spanner	$\tilde{O}(n^{3/2})$	$\tilde{O}(n^{3/4})$	✓
5-spanner	$\tilde{O}(n^{4/3})$	$\tilde{O}(n^{5/6})$	
$O(k^2)$-spanner	$\tilde{O}(n^{1+1/k})$	$\tilde{O}(\Delta^4 n^{2/3})$	✓
$O(n)$ spanner	$o(m)$	$\Omega(\min(\sqrt{n}, n^2/m))$	

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 $O(\Delta)$

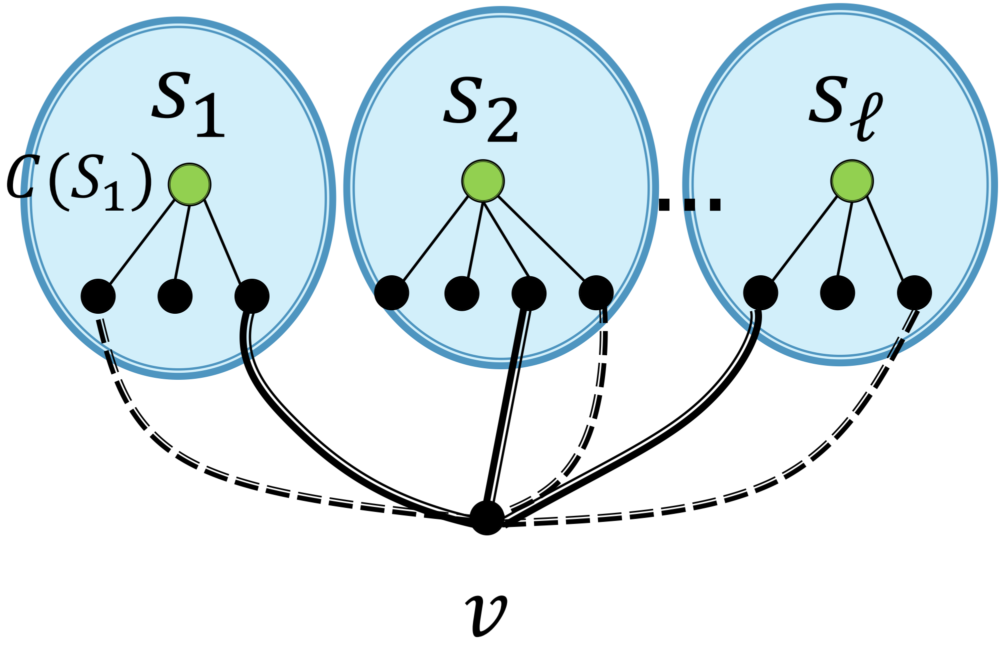


Clustering Alg. for 3-Spanner (Baswana-Sen)

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

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

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

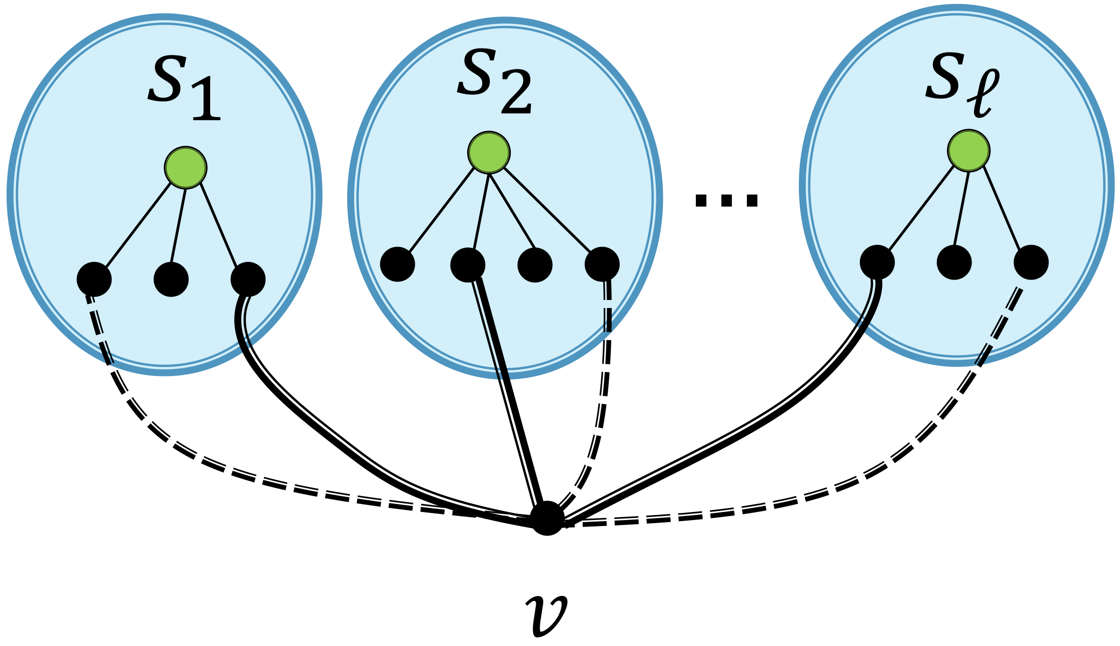


Clustering Alg. for 3-Spanner (Baswana-Sen)

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

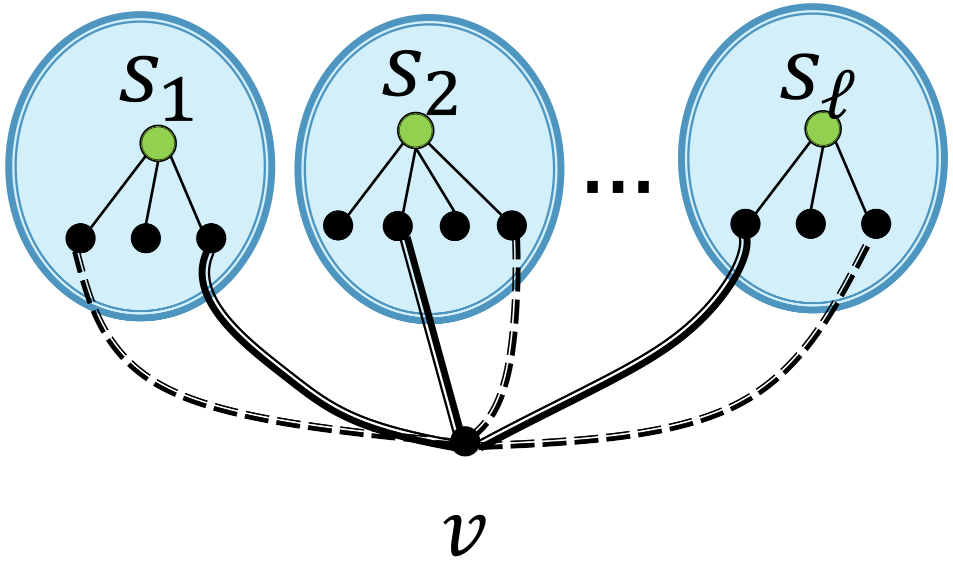
- Picking the first sampled neighbor as a center (for high-deg nodes)
- Picking the first edge to each neighboring star

$N(v)$ $v_1, u_2, \dots v_j, \dots, v_\ell = u, \dots,$

$s(v)$

$N(u)$ $u_1, u_2, \dots u_i, \dots, u_k = v, \dots,$

$s(u)$



By storing poly-log n random bits know if $s \in S$

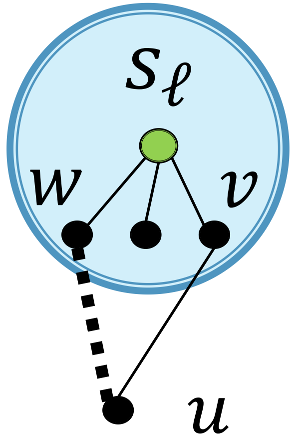
The First Approach

Query: edge (u, v)
Answer: Yes iff $(u, v) \in H$

- If $\mathbf{min}(\mathbf{deg}(u), \boldsymbol{deg}(v)) \leq \sqrt{n}$: "yes"

Remains to handle edges between high-deg vertices

- In $\mathbf{deg(v)}$ probes can compute the center $\boldsymbol{s}(\boldsymbol{v})$
- Should "yes" if $\boldsymbol{v}$ is the first neighbor of $\boldsymbol{u}$ in the cluster of $\boldsymbol{s}(\boldsymbol{v})$



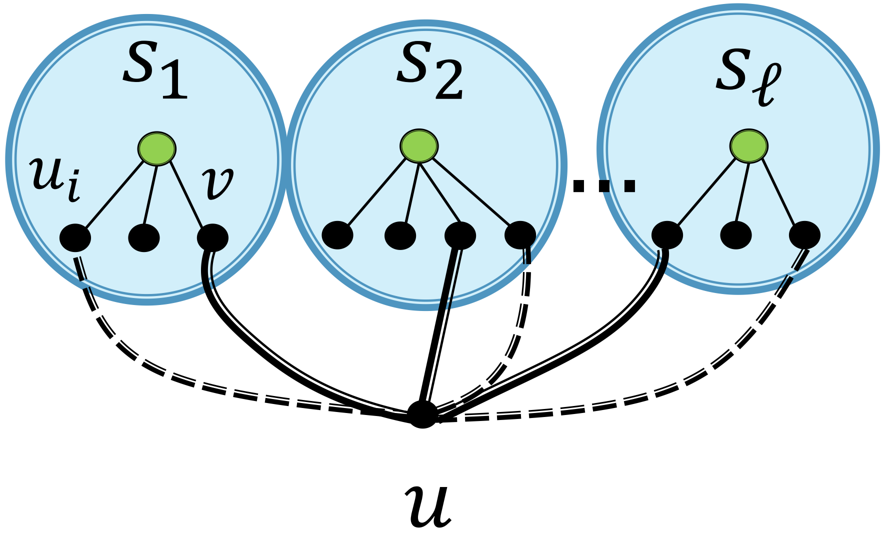
The First Approach

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

$N(u)$: $u_1, u_2, \ldots u_i, \ldots, u_k = v, \ldots,$

$N(v)$: $v_1, u_2, \ldots v_j, \ldots, v_\ell = u, \ldots,$

↑ $s(v)$

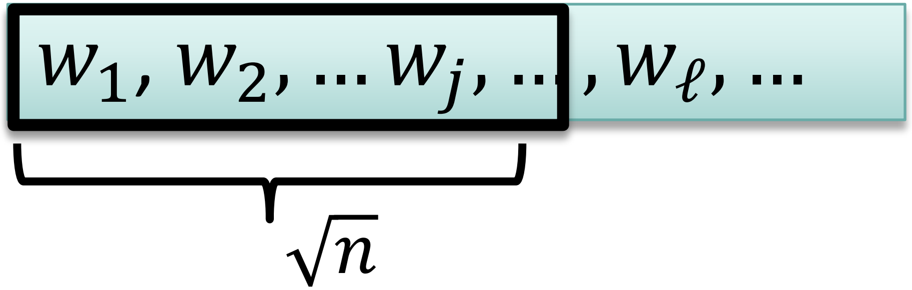


For each $u's$ neighbor u_i
appearing before v in $N(u)$
check if $s(v)$ is the center of u_i

→ Probe complexity $O(\Delta^2)$

The First Multiple Center Approach

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

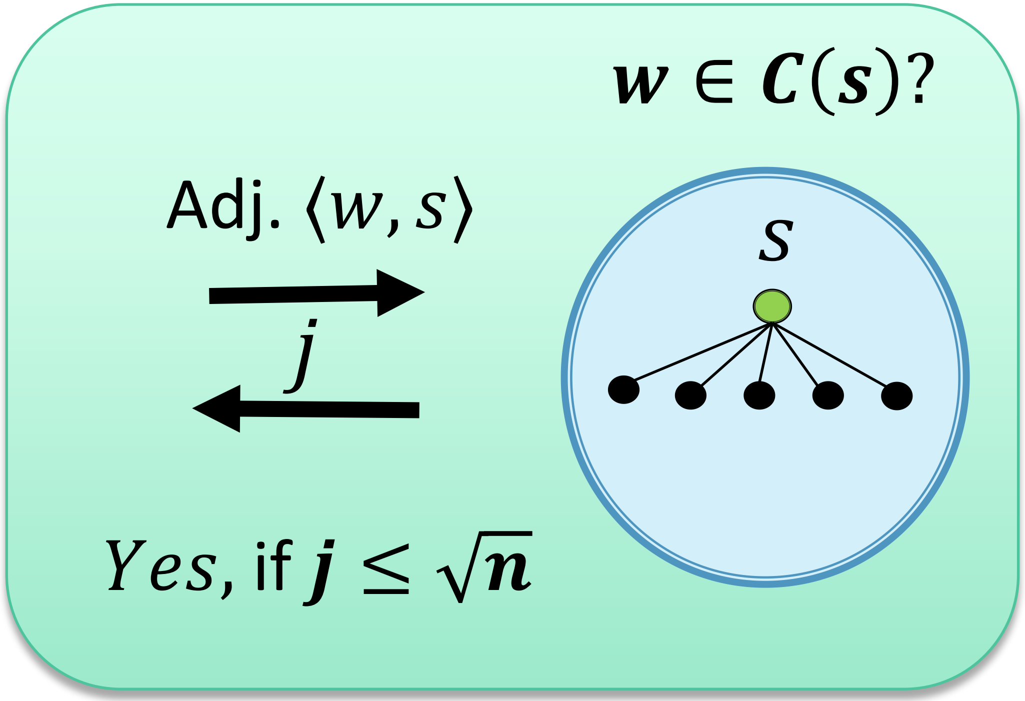


- By Chernoff, w.h.p., high-deg vertex has $\boldsymbol{\Theta(\log n)}$ centers.

- Testing membership to a cluster :

  “does w belong to the cluster of s"?

  Can be done with a **single** adj. probe!



LCA with $\deg(\boldsymbol{u})$ Probes

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

Compute $S(v)$: the first centers of v $\quad v_1, v_2, \dots v_j, \dots, v_\ell, \dots$

$N(u) \quad u_1, u_2, \dots u_i, \dots, u_k = v, u_{k+1} \dots,$

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

For every $\boldsymbol{s} \in \boldsymbol{S}(\boldsymbol{v})$ *and* $\boldsymbol{u_i} < \boldsymbol{v}$:

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

If $\exists \boldsymbol{s'} \in \boldsymbol{S}(\boldsymbol{v})$ for which no $\boldsymbol{u_i} < \boldsymbol{v}$ is in $\boldsymbol{C}(\boldsymbol{s'})$, say "yes"

Probe complexity: $|S(v)| \cdot \deg(u) = \boldsymbol{O}(\boldsymbol{deg}(\boldsymbol{u}) \cdot \log \boldsymbol{n})$

LCA with $\deg(\boldsymbol{u})$ Probes

$N(u)$ $u_1, u_2, \ldots u_i, \ldots, u_k = v, \ldots,$

- **Size analysis:**

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

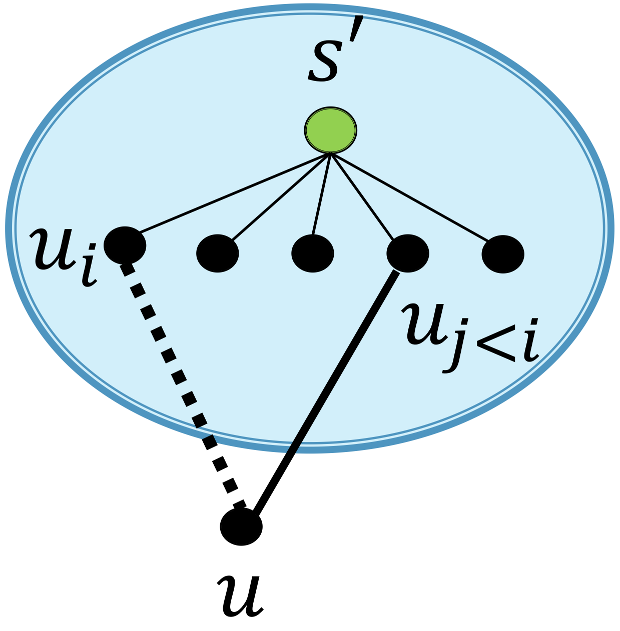
- **Stretch analysis**:

An edge $(u, u_i) \notin H$, with $s' \in S(u_i)$

u_j be the first in $N(u)$ s.t $s' \in S(u_j)$

⇒ $(u, u_j) \in H$

⇒ $dist_H(u, u_i) \leq 3$

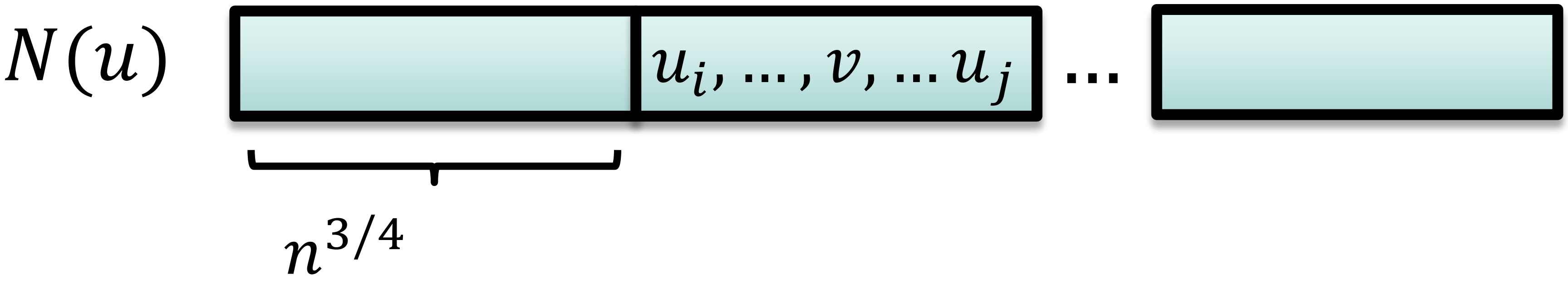


3-Spanner with Sublinear Probe Complexity

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

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

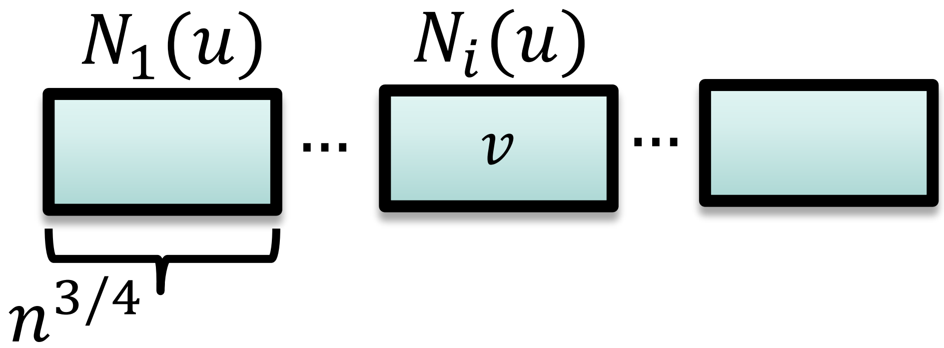
Decompose $N(u)$ into ``independent” blocks of size $\boldsymbol{n}^{3/4}$



Given query $(\boldsymbol{u}, \boldsymbol{v})$ decide based on the block of v in $N(u)$

3-Spanner with Sublinear Probe Complexity

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



Query $(\boldsymbol{u}, \boldsymbol{v})$:

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

Analysis

Query $(\boldsymbol{u}, \boldsymbol{v})$:

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

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

Analysis

Query $(\boldsymbol{u}, \boldsymbol{v})$:

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

Size: In each block $N_i(u)$, add one edge per sampled center.

Overall, $\boldsymbol{|H| = O(n \cdot n^{1/4} \cdot n^{1/4} \cdot log\, n)}$

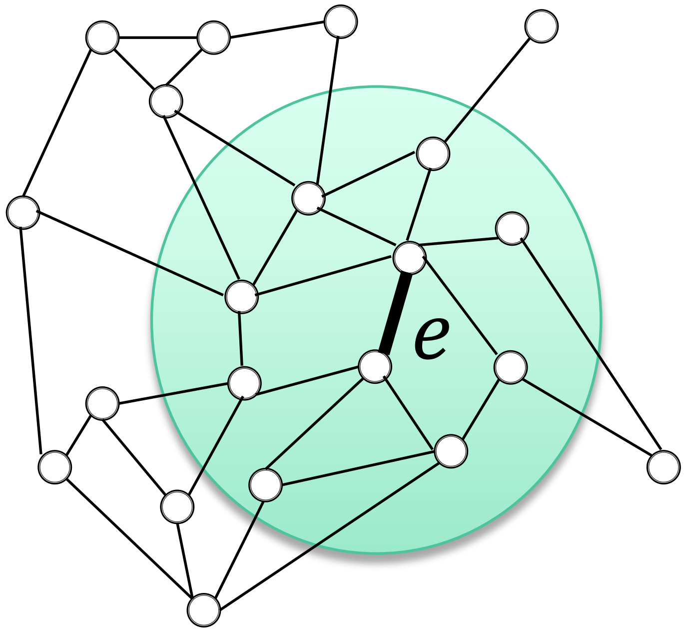
blocks # centers

Spanner LCA with Stretch $O(k^2)$

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

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 $O(\Delta n^{2/3})$



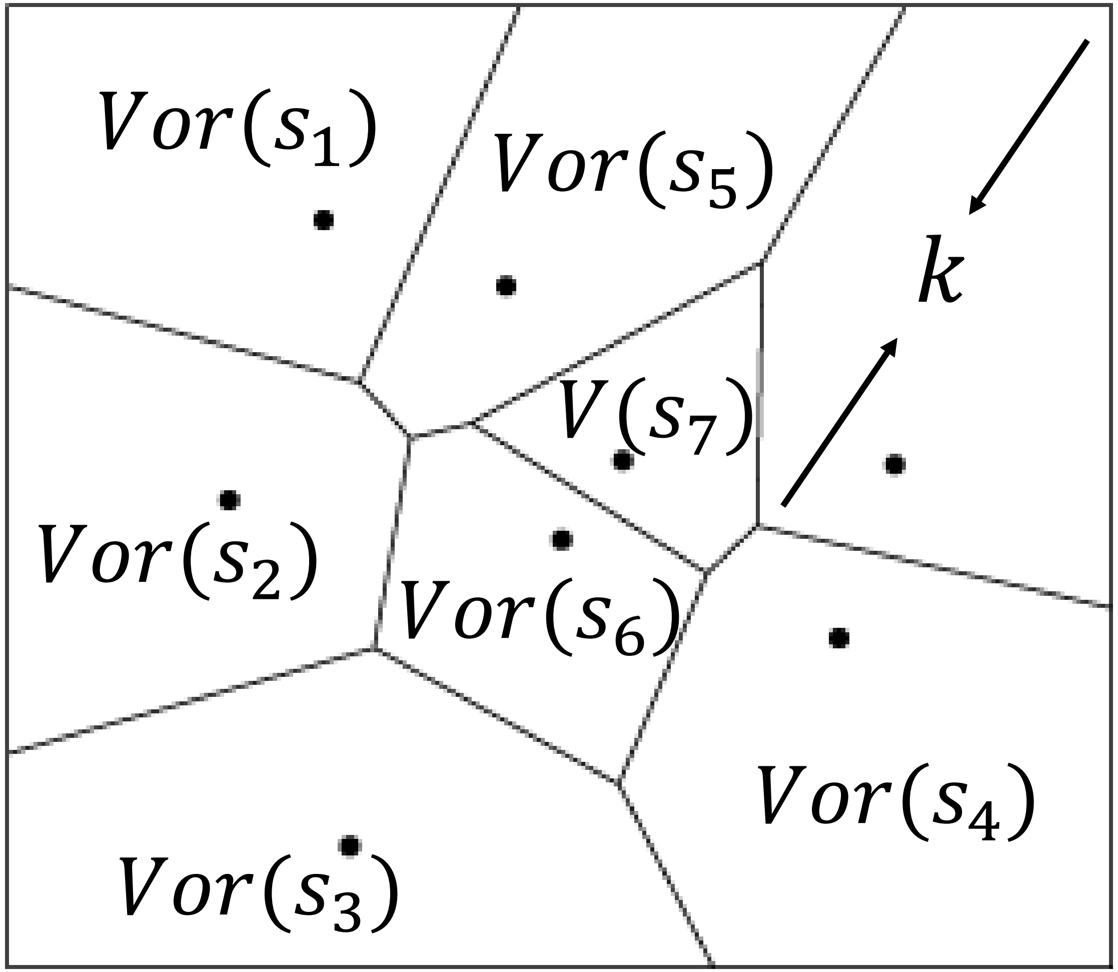
Spanner LCA for the Dense Subgraph

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

Voroni-cells centered at $\boldsymbol{S}$

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

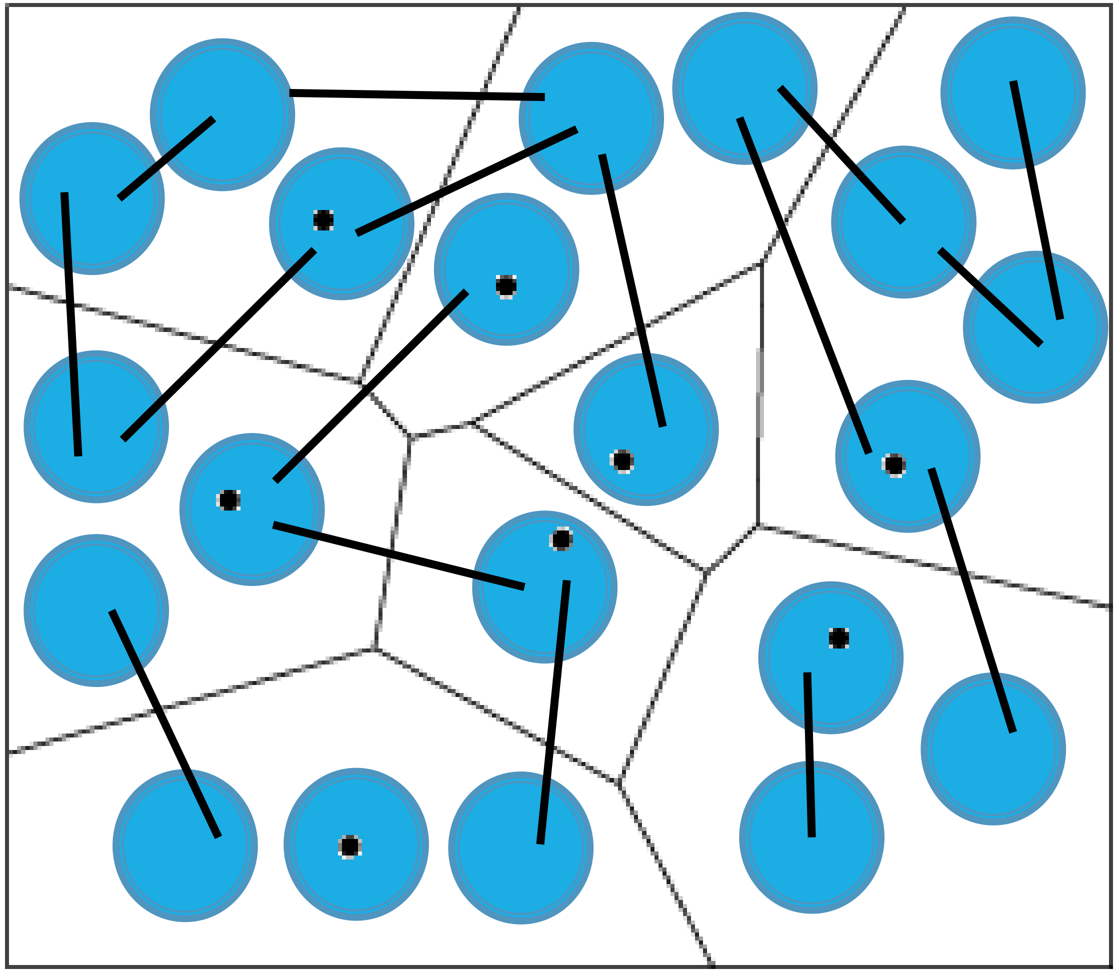
How to connect adjacent cells?



Spanner LCA for the Dense Subgraph

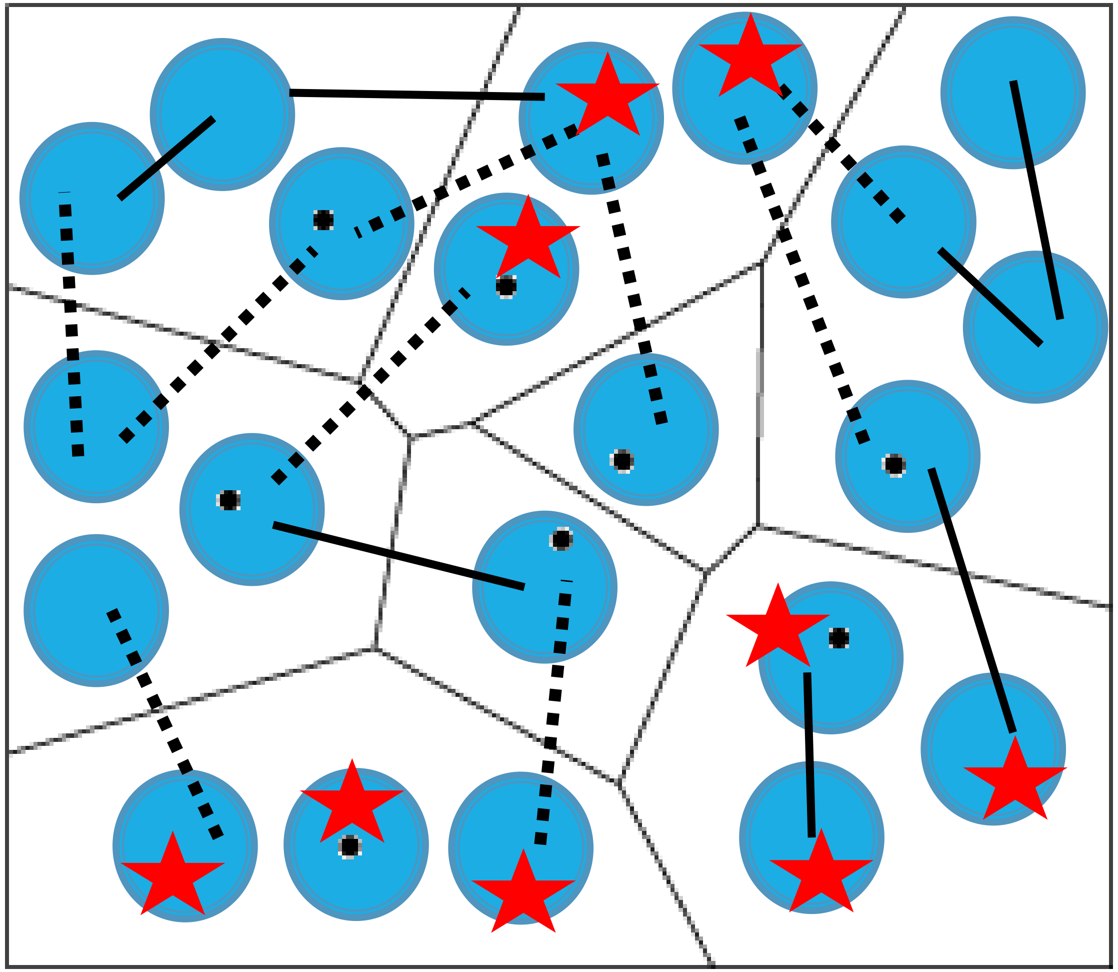
- Subdivide each Voronoi cell into clusters with $\boldsymbol{O(n^{1/3})}$ vertices
- $\boldsymbol{\widetilde{O}(n^{2/3})}$ clusters
- Sample $\boldsymbol{\widetilde{O}(n^{1/3})}$ *centers* $\boldsymbol{S' \subseteq S}$

All clusters of a sampled Voronoi cell are **marked**.



Spanner LCA for the Dense Subgraph

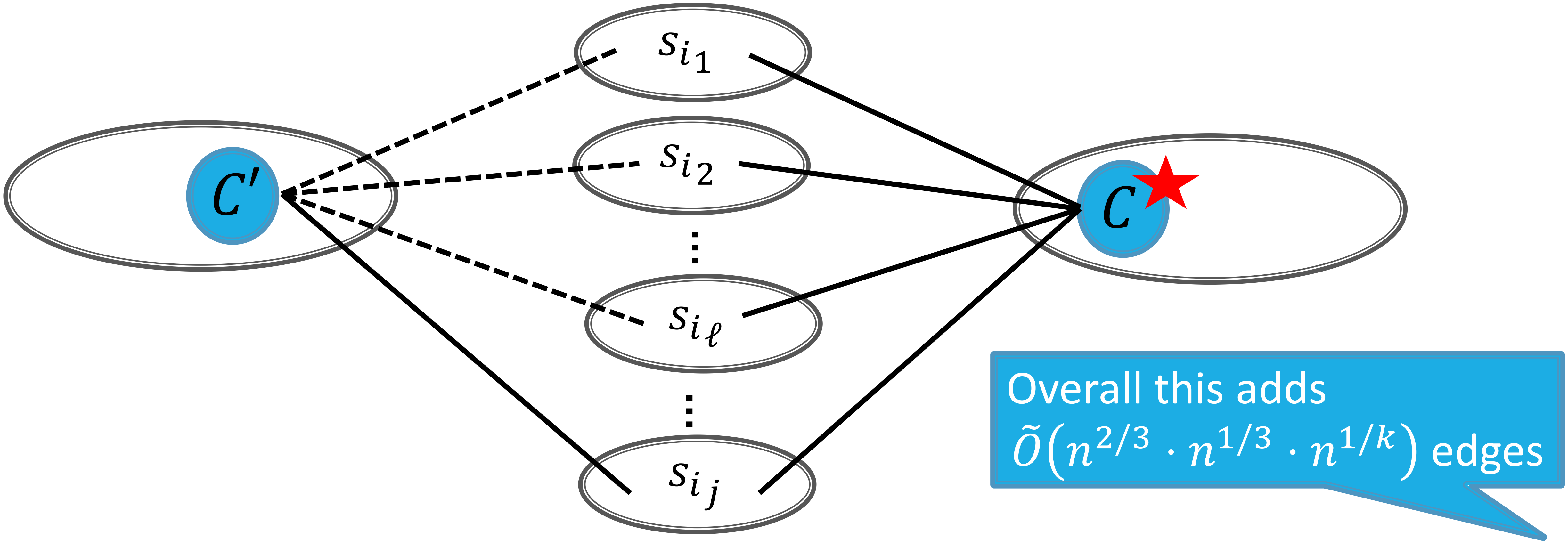
- $\tilde{O}(n^{1/3})$ marked clusters.
- Connect each cluster to all marked neighboring clusters ($\tilde{O}(n)$ edges)
- Connect each cluster with no marked neighbor, to all of its neighbors



Spanner LCA for the Dense Subgraph

For each marked cluster C and cluster C':

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

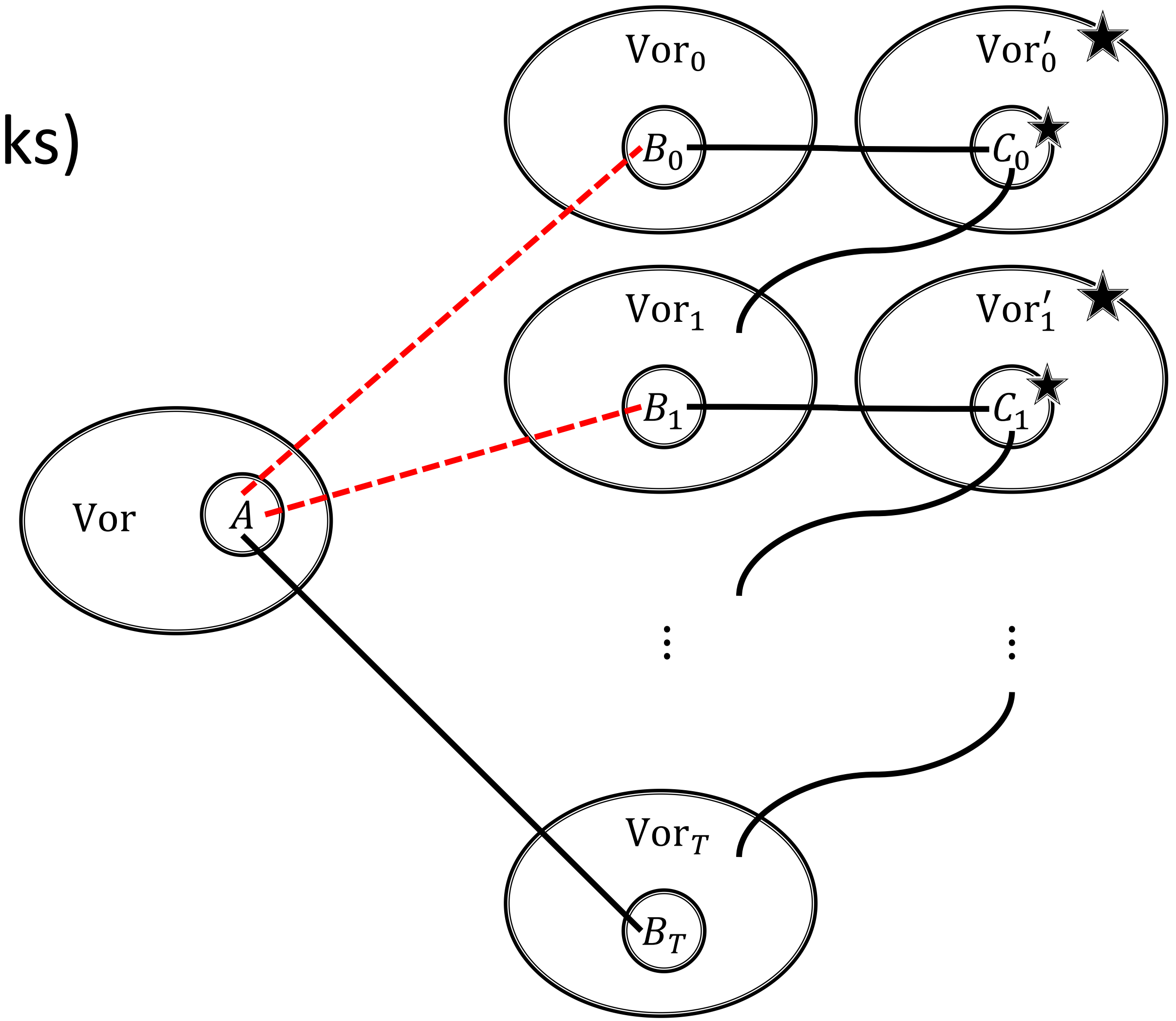


Spanner LCA for the Dense Subgraph

Analysis idea:

Inductive argument (based on random ranks) to show stretch in H_{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

Stretch	# Edges	# probes
$2r-1, r \in \{2,3\}$	$\tilde{O}(n^{1+r})$	$\tilde{O}(n^{1-1/2r})$

Open for $r \geq 4$

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

Improve dependency on Δ

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

Stretch-sensitive lower bounds

Toda!