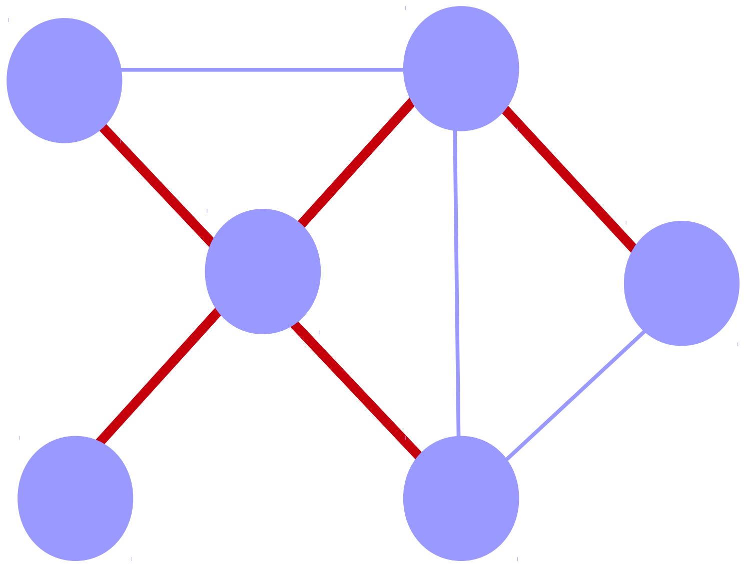
Local Algorithms for Sparse Spanning Graphs

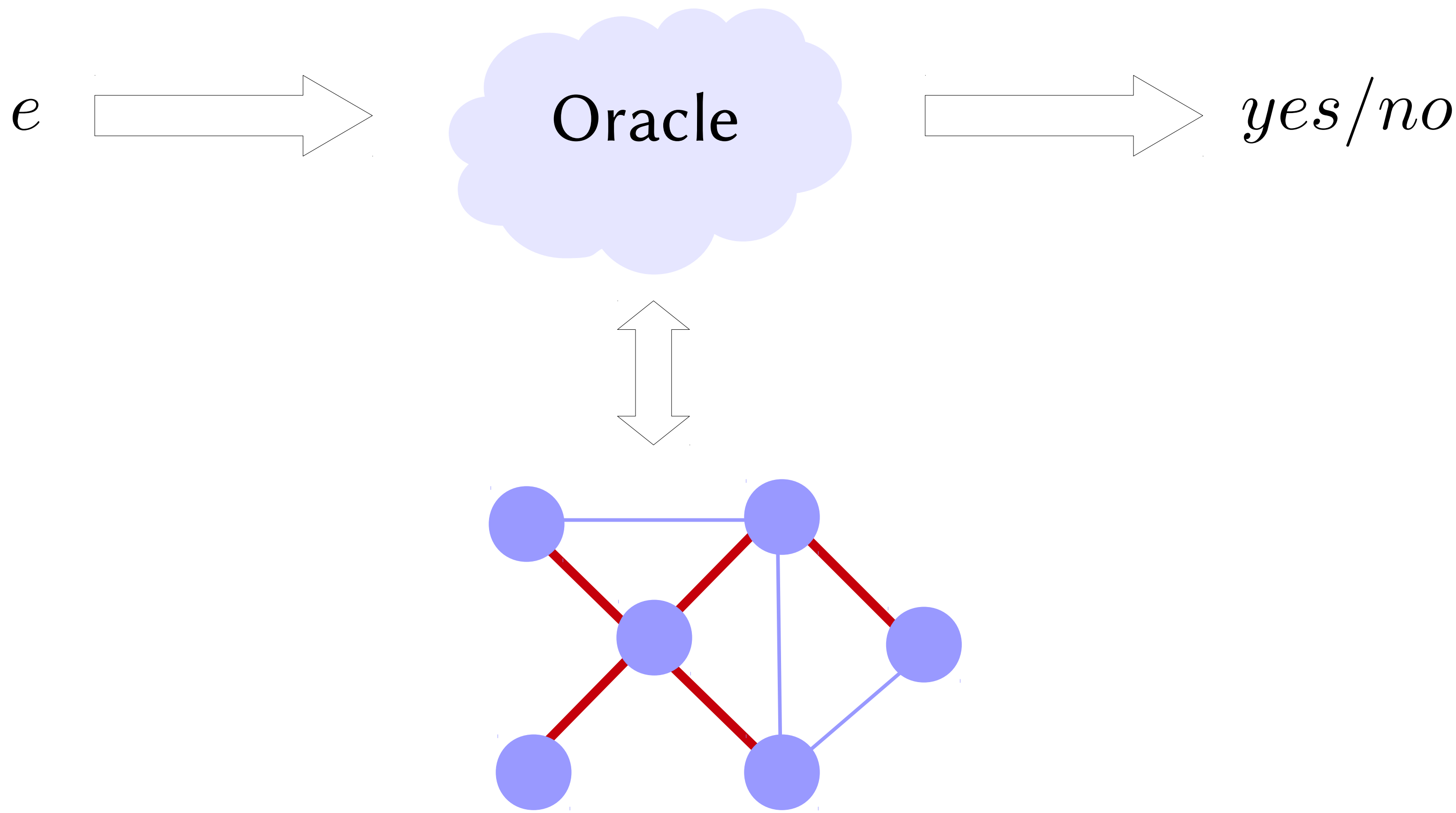
Reut Levi   Dana Ron   Ronitt Rubinfeld

- Intro slides based on a talk given by Reut Levi

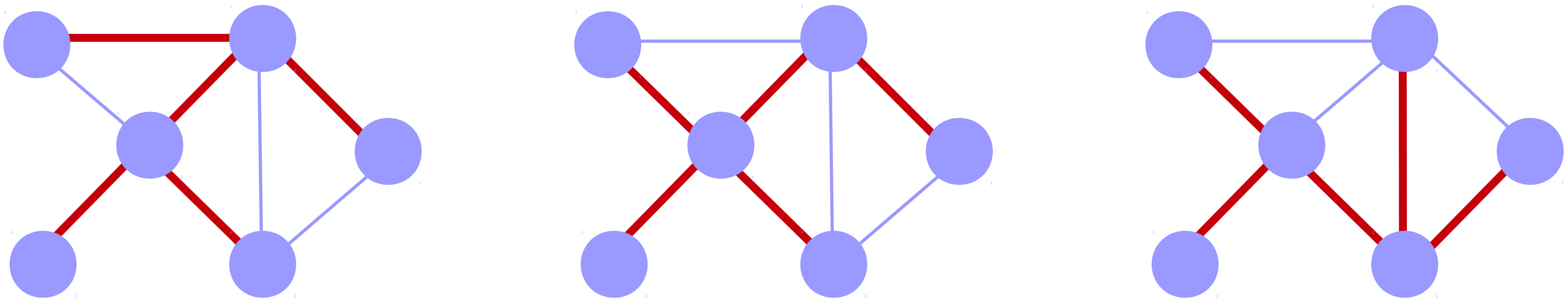
Minimum Spanning Graph
(Spanning Tree)



Local Access to a Minimum Spanning Graph (Spanning Tree)



Local Access to a Minimum Spanning Graph (Spanning Tree)

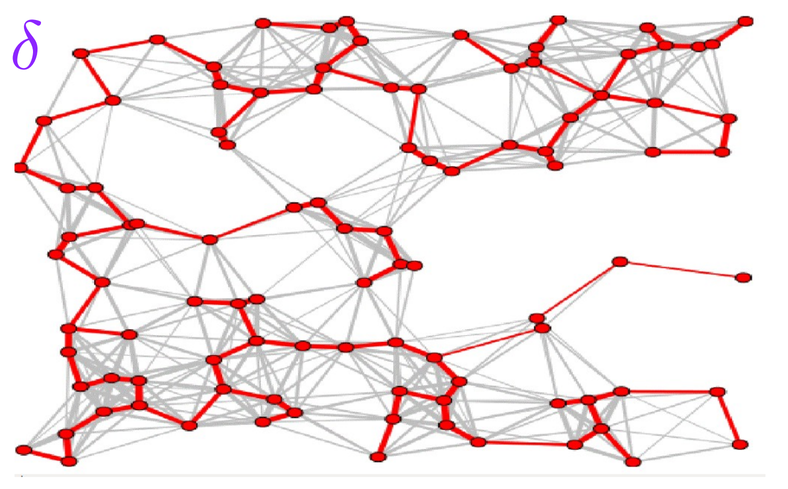


Consistency with the same tree

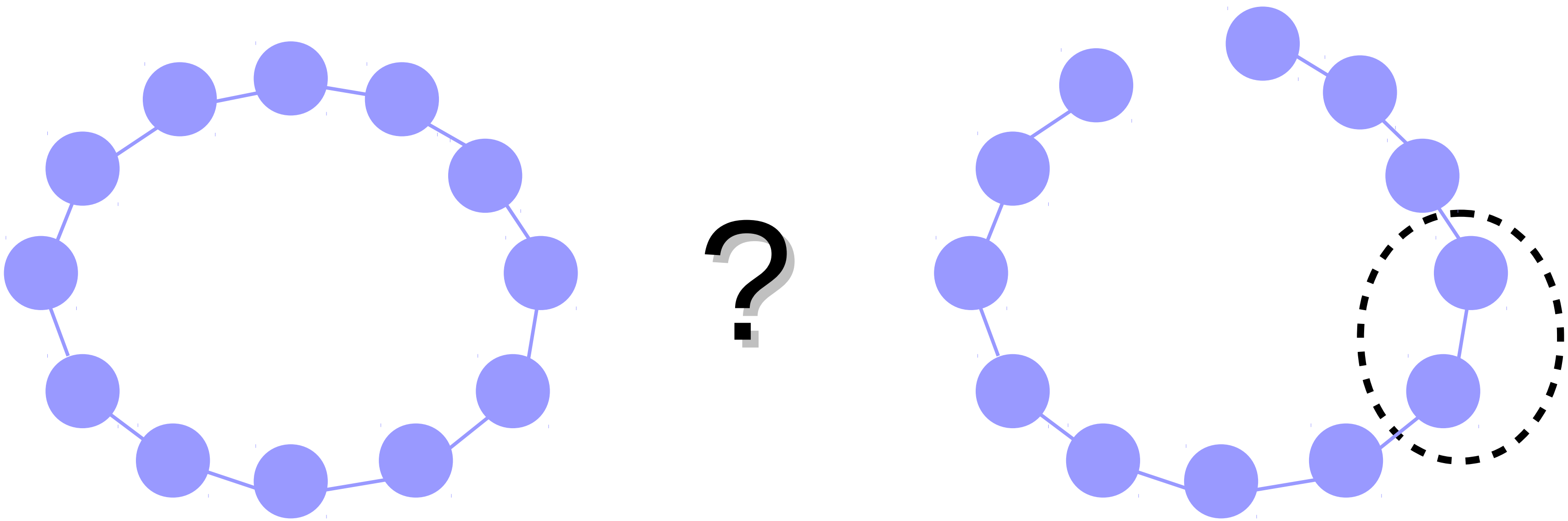
Local Algorithm for Minimum Spanning Graph (Spanning Tree)

Given parameter δ, and query access to $G = (V, E)$ over n vertices and maximum degree d, the algorithm provides oracle access to a subgraph of G, $G' = (V, E')$ such that:

1. G' is connected
2. G' is a tree with probability $\geq 1 - \delta$
3. G' is determined by G and internal randomness

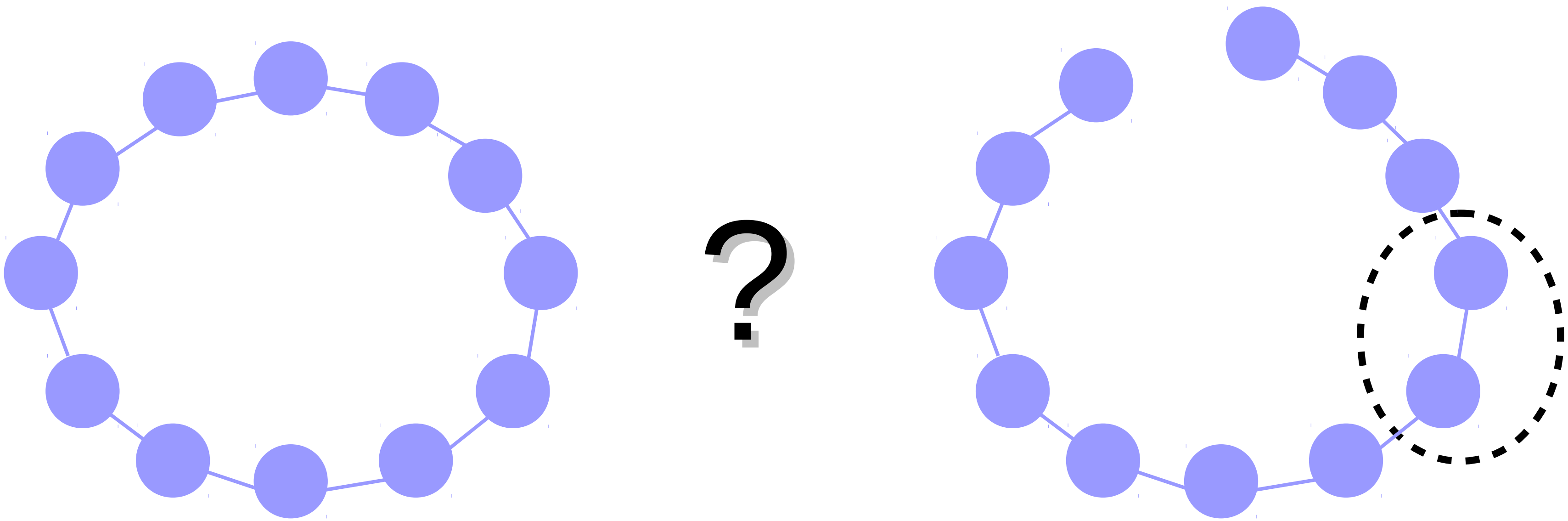


Local Algorithm for Minimum Spanning Graph (Spanning Tree)



YES on every edge

Local Algorithm for Minimum Spanning Graph (Spanning Tree)

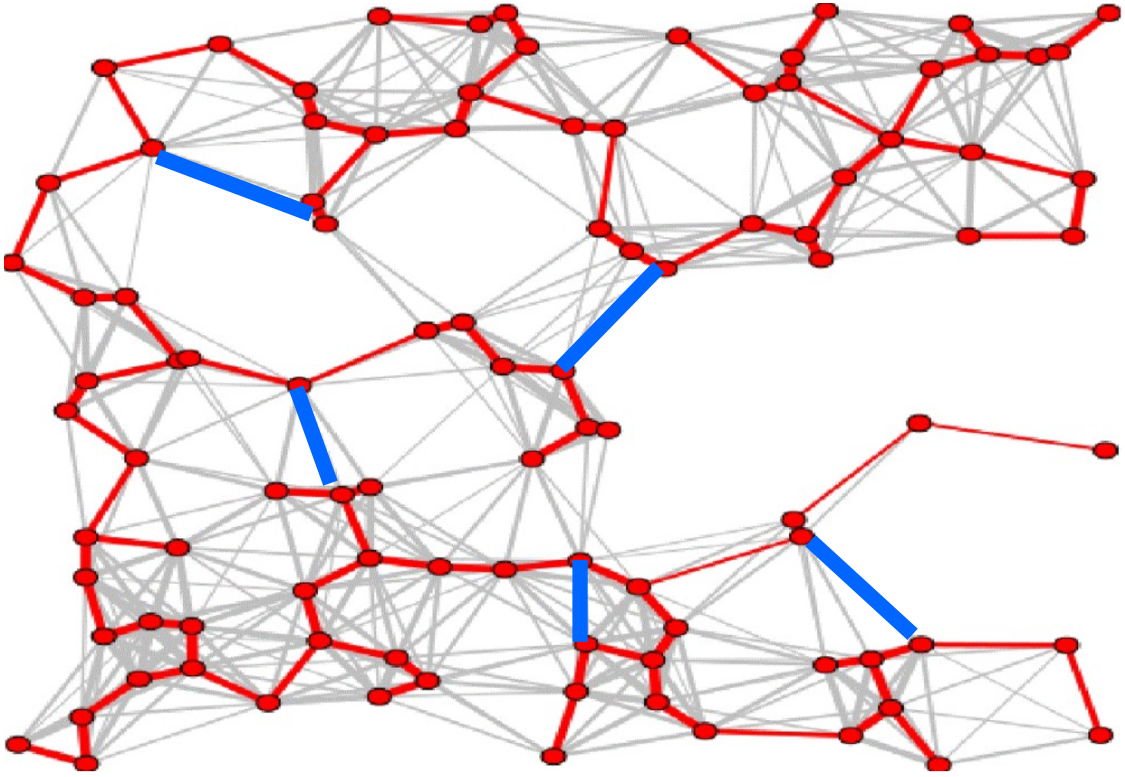


YES on every edge

➔Requires $\Omega(n)$ samples

A relaxation.

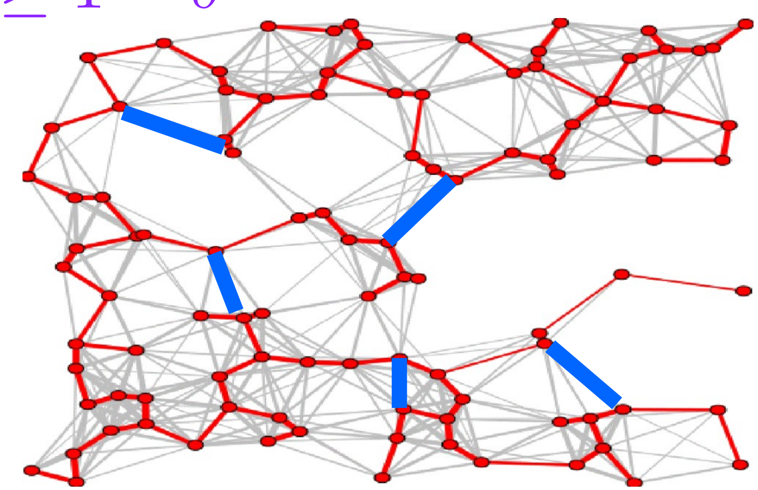
Allow a few extra edges



Local Algorithm for Sparse Spanning Graph

Given parameters δ, ϵ, and query access to $G = (V, E)$ over n vertices and maximum degree d, the algorithm provides oracle access to a subgraph of G, $G' = (V, E')$ such that:

1. G' is connected
2. $|E'| \leq (1 + \epsilon)|V|$ with probability $\geq 1 - \delta$
3. G' is determined by G and internal randomness



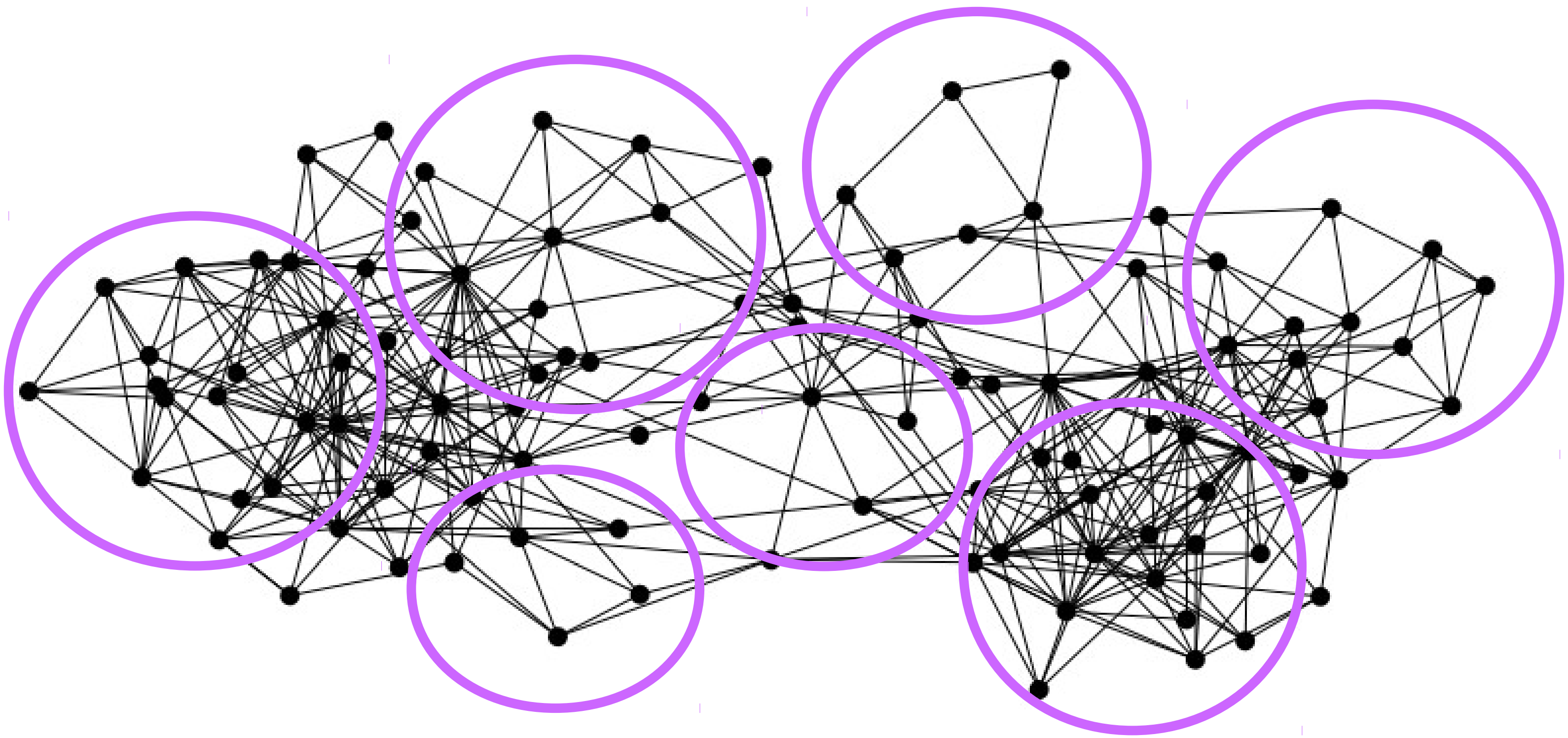
The Algorithm

- We will present a sparse spanning graph algorithm for **High Expansion** Graphs
  - Phase 1: Global algorithm
  - Phase 2: Local algorithm

Expander Graphs

- Expanders are **sparse**, yet **highly connected** graphs.
- We define expansion as follows:
    - For a subset $S \subset V, \partial(S)$ is the set of vertices from $V \backslash S$ with a neighbor in S.
    - A graph is an (s, α) vertex-expander if for all sets S of size at most s, $\alpha \leq \frac{|\partial(S)|}{|S|}$

Sparse Spanning Graph Algorithm for High Expansion Graphs



Global Algorithm

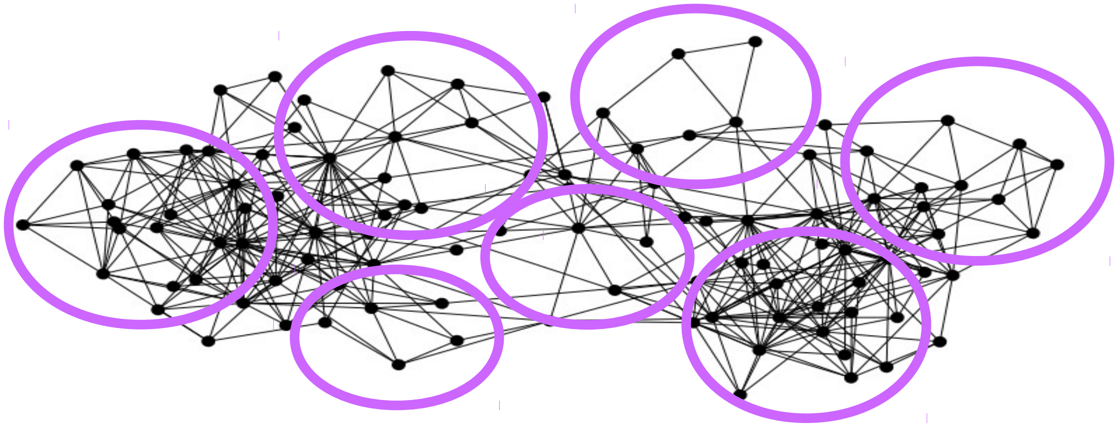
Given a parameter k, we partition the graph (or some of it):

- Randomly sample $\sqrt{\frac{\epsilon n}{2}}$ vertices, denoted as **Centers**
- For each vertex we want to find its **closest center** (up to k)
- How? **BFS** of depth k from each center, breaking ties by id

Global Algorithm

The edges of the spanning graph are:

- Edges of the BFS-tree over closest vertices of a center
- Edges of vertices not close enough to any center (singletons)
- Single edges connecting adjacent components



Choosing the radius k

- $s = \sqrt{\frac{2n}{\epsilon} log(n/\delta)}$
- $\Gamma_k(v)$ is the k neighborhood of v
- $k^* = min_k\{|\{v : |\Gamma_k(v)| \geq s\}| \geq (1 - \frac{\epsilon}{2d})|V|\}$

➔ k^* is the minimum distance needed for most vertices to see enough vertices in their neighborhood.

Correctness

- G' is sparse with probability $\geq 1 - \delta$
- G' is connected

Correctness: G' is Sparse

- Within centers: all BFS-trees of all centers contains $\leq n$ edges.
- Between centers: edges added between pairs of centers are $\leq \frac{\epsilon n}{2}$
- Singletons: $\leq \frac{\epsilon n}{2d}$, each has $\leq d$ edges, in total $\leq \frac{\epsilon n}{2}$ edges.

Correctness: G' is Sparse

- Within centers: all BFS-trees of all centers contains $\leq n$ edges.
- Between centers: edges added between pairs of centers are $\leq \frac{\epsilon n}{2}$
- Singletons: $\leq \frac{\epsilon n}{2d}$, each has $\leq d$ edges, in total $\leq \frac{\epsilon n}{2}$ edges.

➔ The spanning graph has $\leq (1+\epsilon)n$ edges.

Correctness: G' is Sparse

- For each v for which $|\Gamma_k(v)| \geq s$ the probability that no vertex in $\Gamma_k(v)$ is selected as a center is

$$\leq (1 - \frac{s}{n})^{\sqrt{n\epsilon/2}} = (1 - \frac{log(n/\delta)}{\sqrt{n\epsilon/2}})^{\sqrt{n\epsilon/2}} < \delta/n$$

Correctness: G' is Sparse

- For each v for which $|\Gamma_k(v)| \geq s$ the probability that no vertex in $\Gamma_k(v)$ is selected as a center is

$$\leq (1 - \frac{s}{n})^{\sqrt{n\epsilon/2}} = (1 - \frac{log(n/\delta)}{\sqrt{n\epsilon/2}})^{\sqrt{n\epsilon/2}} < \delta/n$$

- From union bound, every such vertex is close enough to some center with probability $\geq 1 - \delta$

Correctness: G' is Sparse

- For each v for which $|\Gamma_k(v)| \geq s$ the probability that no vertex in $\Gamma_k(v)$ is selected as a center is

$$\leq (1 - \frac{s}{n})^{\sqrt{n\epsilon/2}} = (1 - \frac{log(n/\delta)}{\sqrt{n\epsilon/2}})^{\sqrt{n\epsilon/2}} < \delta/n$$

- From union bound, every such vertex is close enough to some center with probability $\geq 1 - \delta$
- By choosing $k^* = min_k\{|\{v : |\Gamma_k(v)| \geq s\}| \geq (1 - \frac{\epsilon}{2d})|V|\}$, only a small fraction will be singletons, with probability $\geq 1 - \delta$

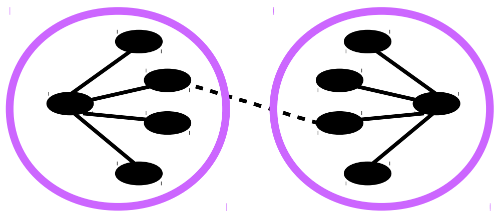
Correctness: G' is Connected

- It's enough to show that each 2 centers in G' are connected.
- Why? Each vertex has a path to some center:
  - Each vertex assigned to a center, has a path to a center
  - Each vertex not assigned to any center, has all its edges in E'

Correctness: G' is Connected

Each 2 centers in G' are connected:

1. All vertices of the shortest path are assigned to those centers.

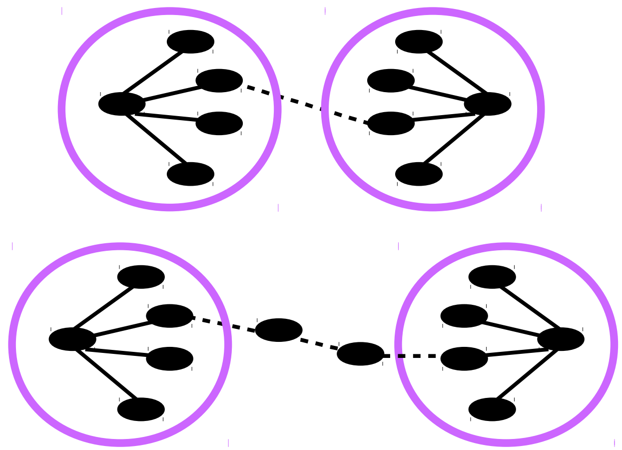


Correctness: G' is Connected

Each 2 centers in G' are connected:

I. All vertices of the shortest path are assigned to those centers.

II. All vertices are either assigned to those centers, or singletons.



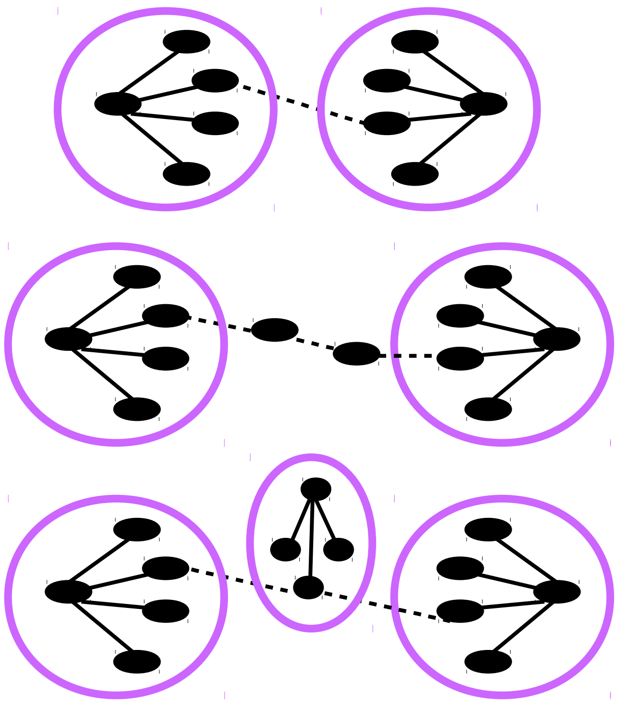
Correctness: G' is Connected

Each 2 centers in G' are connected:

I. All vertices of the shortest path are assigned to those centers.

II. All vertices are either assigned to those centers, or singletons.

III. Some vertex on the path is assigned to another center.



Local Algorithm

- Randomly sample $\sqrt{\frac{\epsilon n}{2}}$ centers.
- Given an edge (u, v), perform BFS to depth k from both u and v

Local Algorithm

- Randomly sample $\sqrt{\frac{\epsilon n}{2}}$ centers.
- Given an edge (u, v), perform BFS to depth k from both u and v

I. If either u or v are singletons, return YES.

II. If they share a center, run min-id BFS from the center.

III. If the centers are different, check their min-id shortest path.

Local Algorithm Complexity

- Let n_k be the maximum size of a k neighborhood
- The query and time complexity is $O(d \cdot n_k)$

Local Algorithm Complexity

- Let n_k be the maximum size of a k neighborhood
- The query and time complexity is $O(d \cdot n_k)$
- $k^* = min_k\{|\{v : |\Gamma_k(v)| \geq s\}| \geq (1 - \frac{\epsilon}{2d})|V|\}$
- $k^{*\prime} = min_k\{|\{v : |\Gamma_k(v)| \geq s\}| \geq (1 - \frac{\epsilon}{4d})|V|\}$
- Estimate $k^* \leq k \leq k^{*\prime}$ with probability $\geq 1 - \delta$

Estimating the radius k

- Randomly sample $r = \Theta(1/\epsilon^2 log(1/\delta))$ vertices
- For each v_i from $v_1, ..., v_r$, set $k_i = min_k\{\Gamma_k(v_i) \geq s\}$
- Assume $k_1 \leq ... \leq k_r$, set $\hat{k} = k_{(1-\frac{3\epsilon}{8d})r}$
- By Chernoff, $k^* \leq k \leq k^{*\prime}$ with probability $\geq 1 - \delta$

Local Algorithm Complexity

- Since G is (s, α) expander, we can bound $k^{*\prime}$ using s and α
- We can now bound $O(d \cdot n_k)$
- We obtain the complexity $d \cdot s^{log_\alpha d}$ (where $s = \sqrt{\frac{2n}{\epsilon} log(n/\delta)}$)
- For high expansion graphs we get nearly $O(\sqrt{n})$

Conclusion

- Local algorithm for sparse spanning graph of expanders
- Other graph families? General graphs?