On Sparse Spanners

—

Construction locale de sous-graphes couvrants épars

Cyril Gavoille

Université de Bordeaux, France

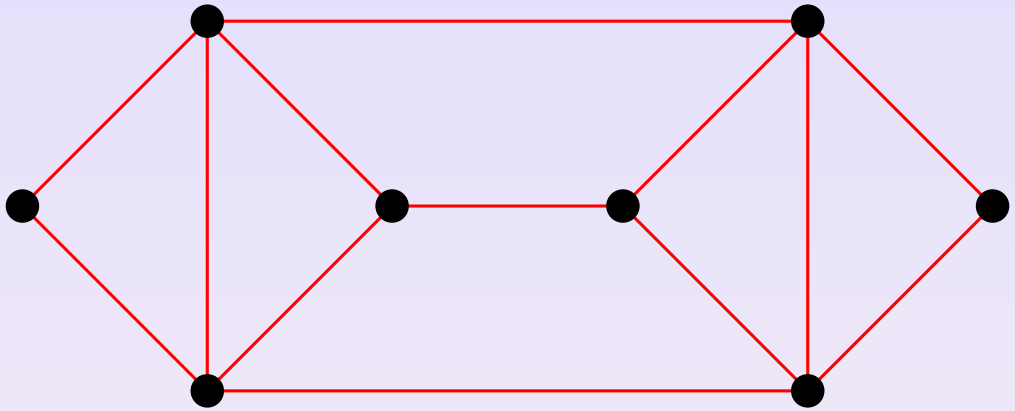
29-30 mai, 2008, La Rochelle

Outline

Outline

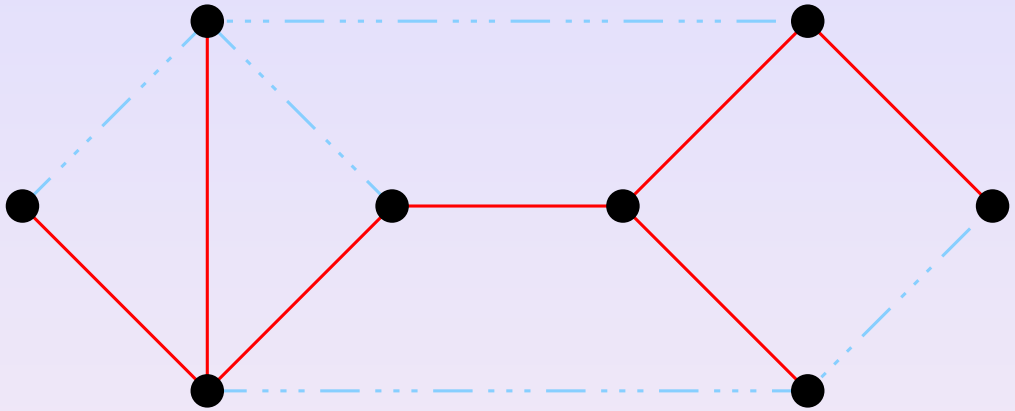
What is a Spanner?

A **spanner** of a graph G is a subgraph spanning $V(G)$



What is a Spanner?

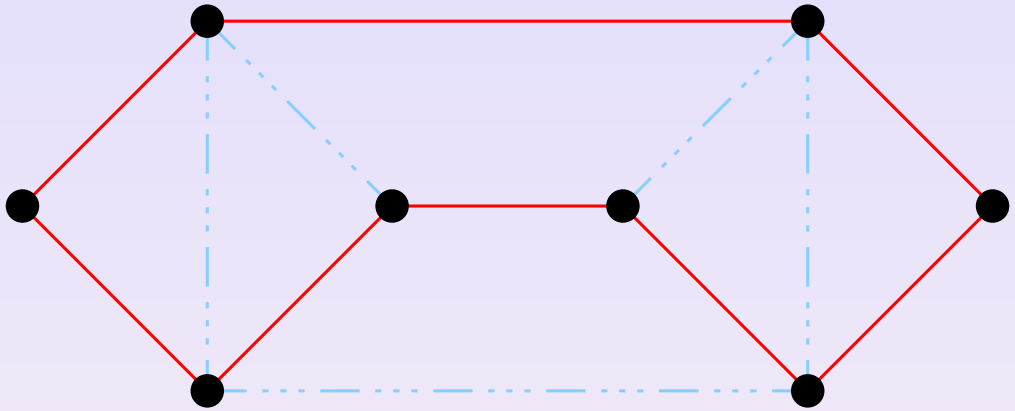
A **spanner** of a graph G is a subgraph spanning $V(G)$



- a spanning tree

What is a Spanner?

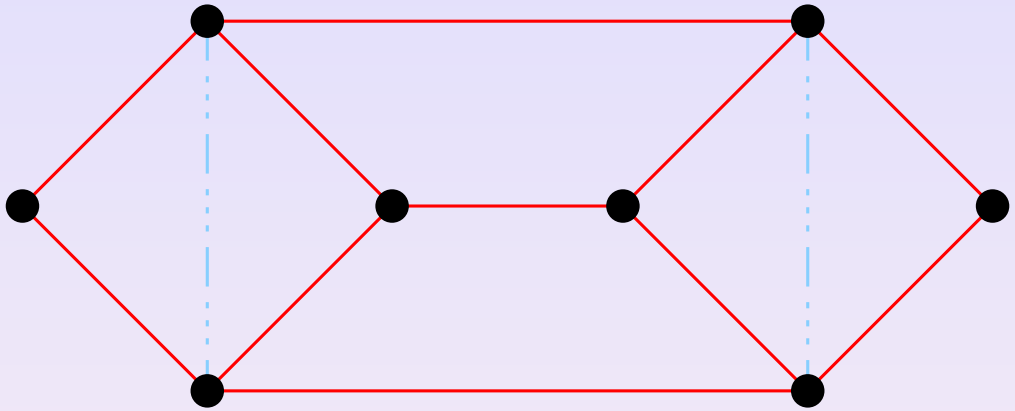
A **spanner** of a graph G is a subgraph spanning $V(G)$



- a spanning tree
- a Hamiltonian cycle

What is a Spanner?

A **spanner** of a graph G is a subgraph spanning $V(G)$



- a spanning tree
- a Hamiltonian cycle
- a maximal bipartite subgraph
- ...

Approximate Distance Spanners

There are two "natural" criteria for a spanner of G:

- **size:** its number of edges.
- **stretch:** its maximum distance distortion from G.

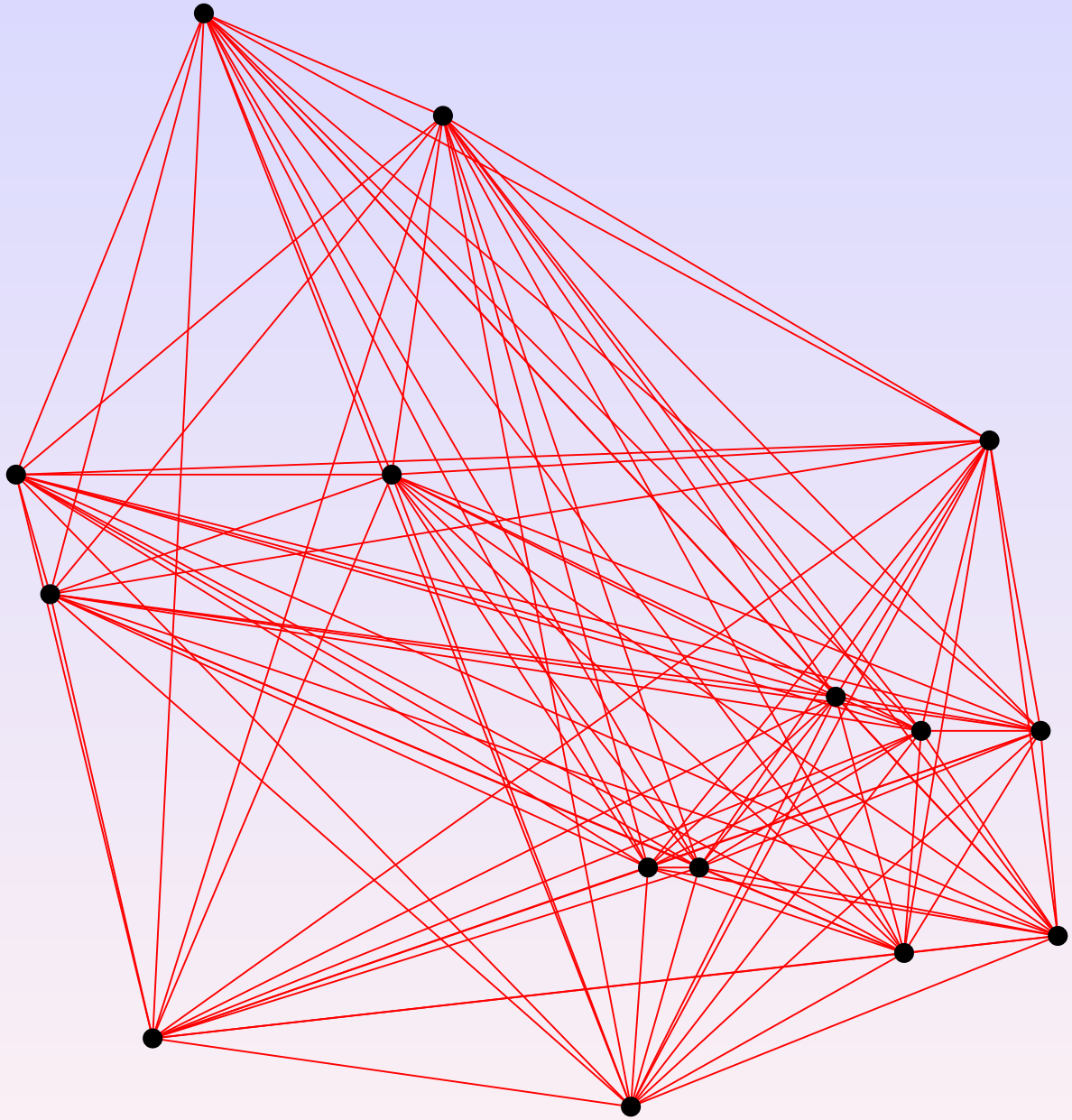
Approximate Distance Spanners

There are two "natural" criteria for a spanner of G:

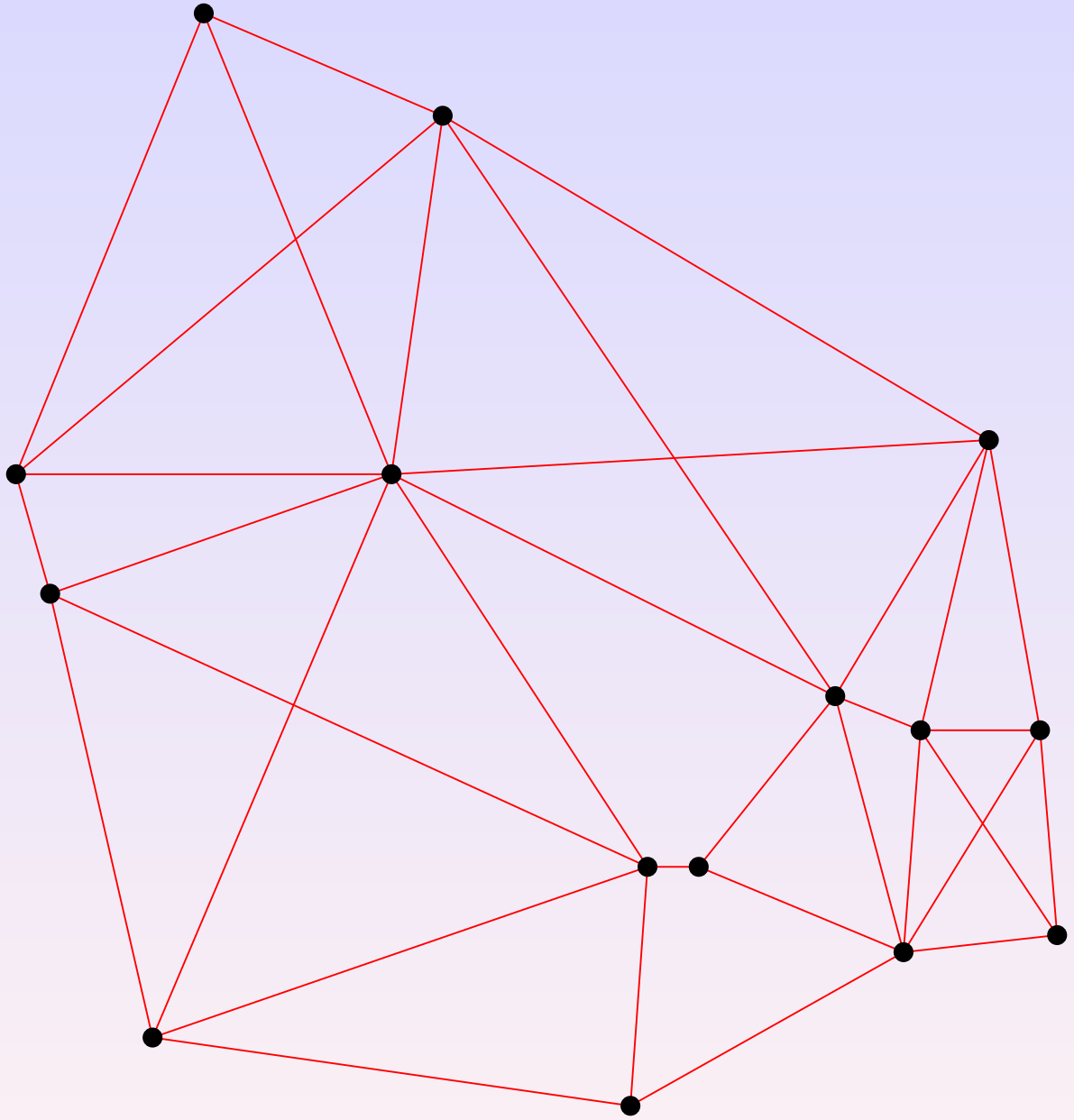
- **size:** its number of edges.
- **stretch:** its maximum distance distortion from G.

Goals:

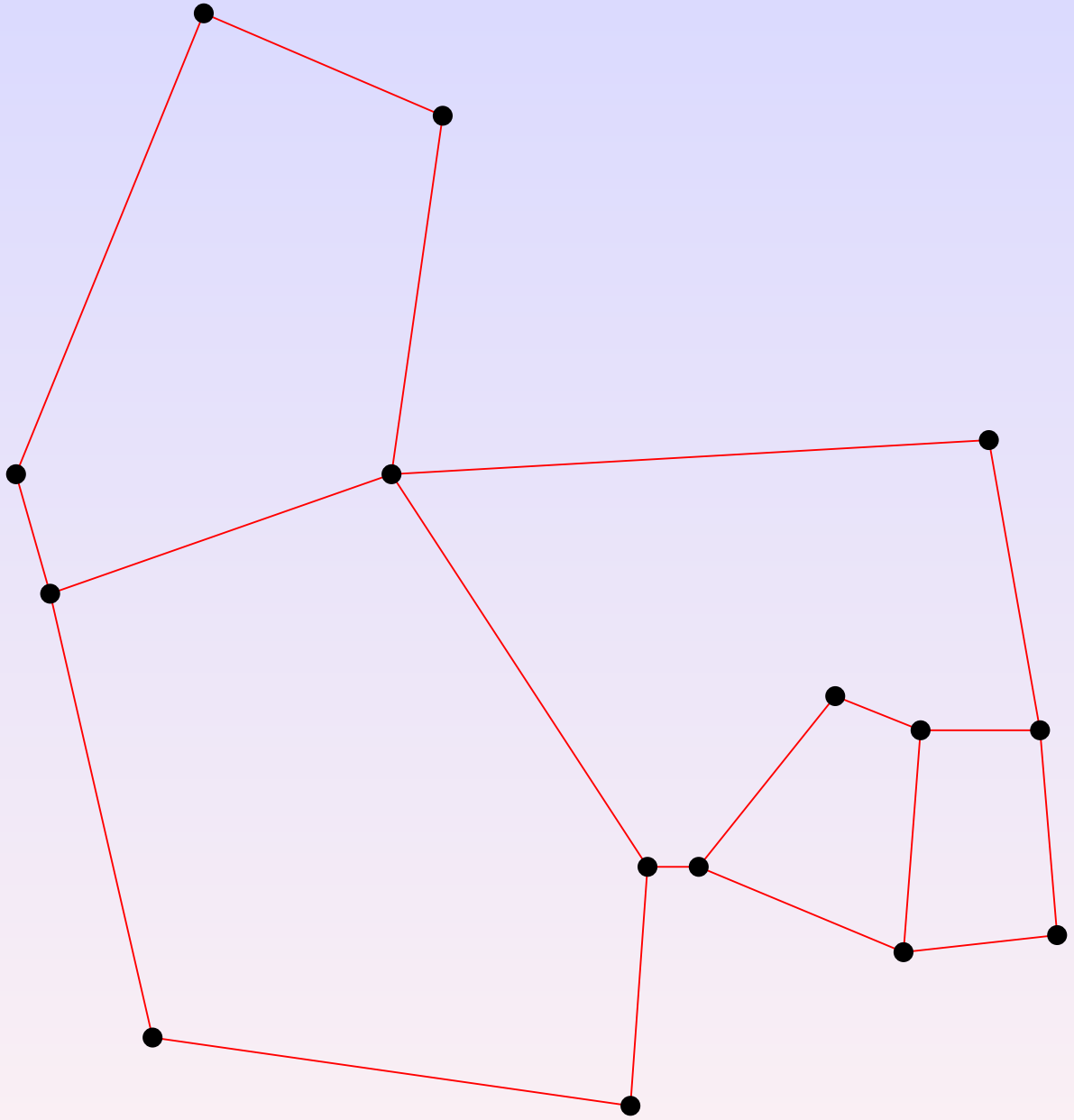
- find a good skeleton of the graph;
- decrease the size of the graph while preserving distances;
- optimize stretch-size tradeoffs.



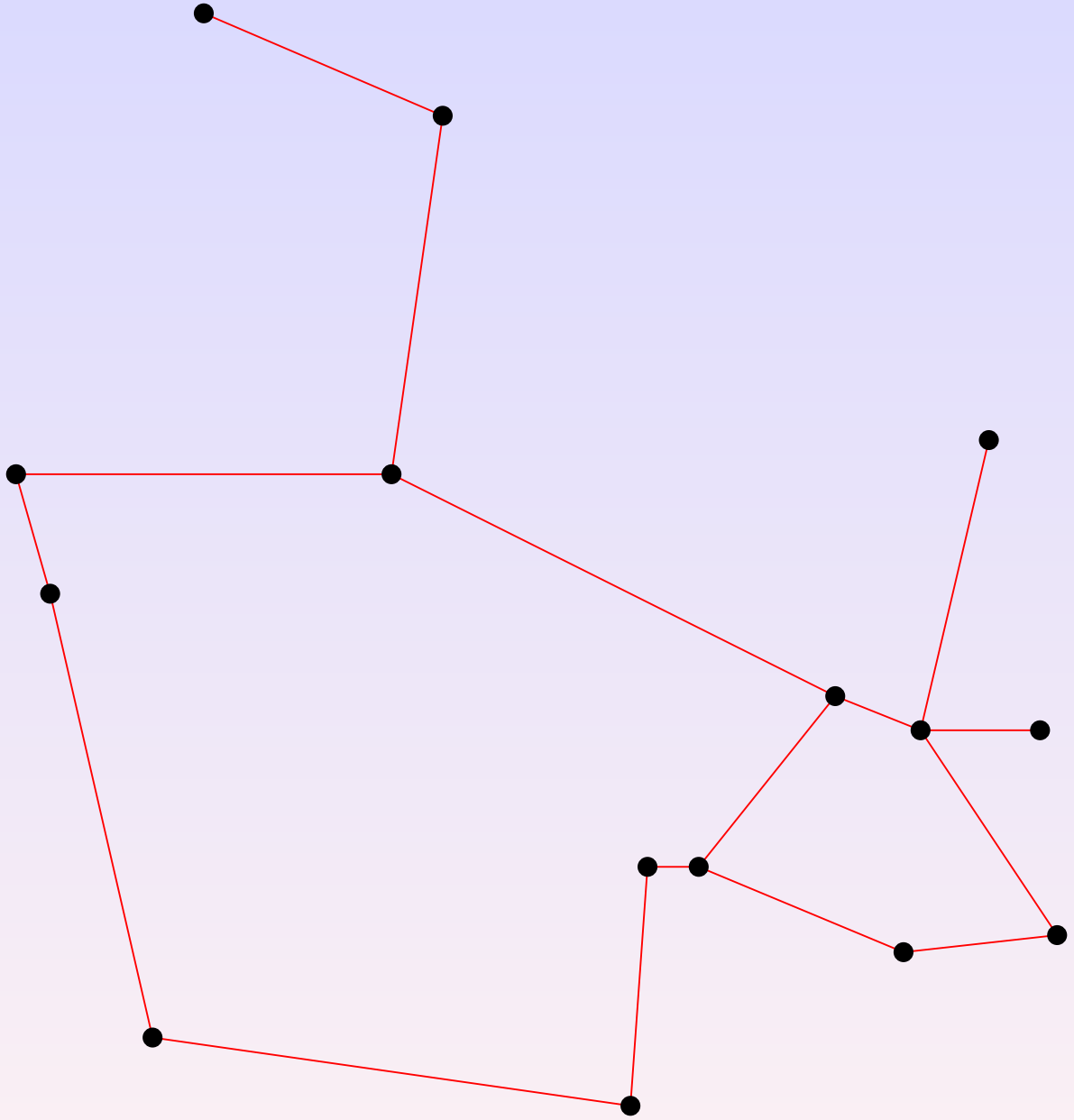
A complete Euclidian graph on 15 nodes



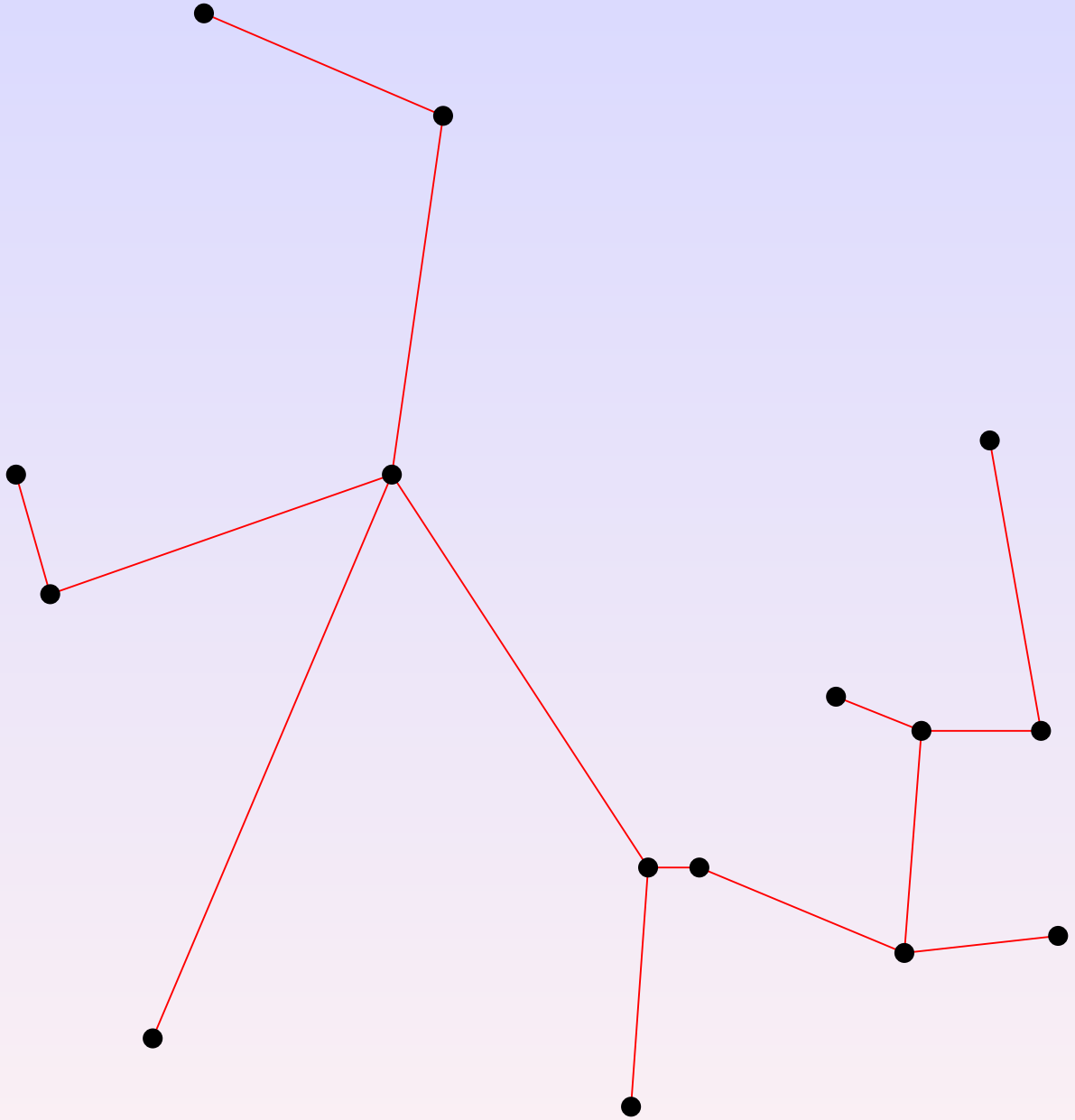
A (minimum cost) spanner with stretch 1.2



A (minimum cost) spanner with stretch 1.7



A (minimum cost) spanner with stretch 2.0



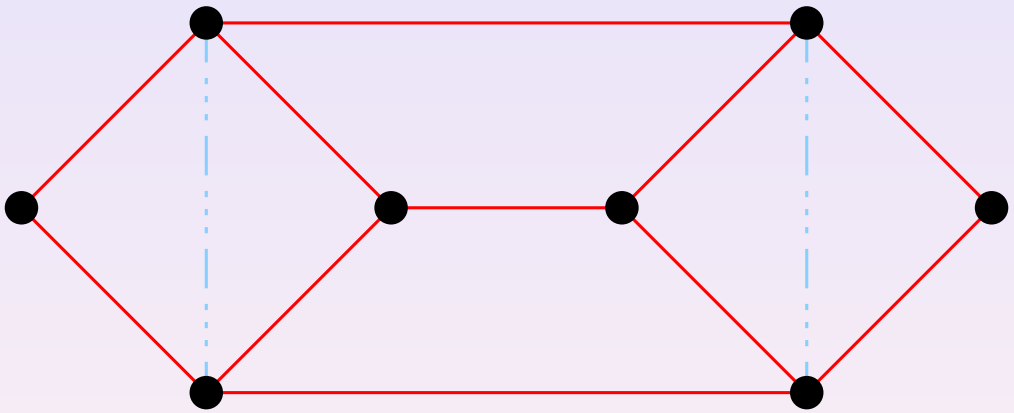
A (minimum cost) spanner with stretch 3.0

Definition

Definition

An (α, β)-spanner S of G is spanner of G satisfying $d_S(x, y) \leqslant \alpha \cdot d_G(x, y) + \beta$ for all $x, y \in V(G)$.

A $(2, 0)$-spanner of size 11 which is $(1, 1)$-spanner as well.

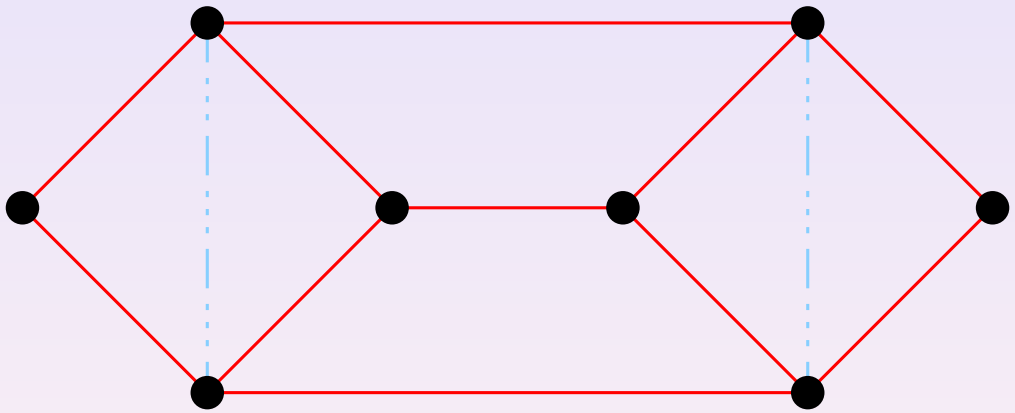


Definition

Definition

An (α, β)-spanner S of G is spanner of G satisfying $d_S(x, y) \leqslant \alpha \cdot d_G(x, y) + \beta$ for all $x, y \in V(G)$.

A $(2, 0)$-spanner of size 11 which is $(1, 1)$-spanner as well.



$$\text{stretch}(S) = \max_{(x,y)\in E(S)} \text{stretch}(x, y) = (\alpha, \beta)$$

Spanners to do What?

Formally introduced by [Peleg-Ullman '87]: *"An optimal synchronizer for the Hypercube"* (440 Google hits)

Used for:

- communication networks
- distributed systems
- network design

Spanners to do What?

Formally introduced by [Peleg-Ullman '87]: *"An optimal synchronizer for the Hypercube"* (440 Google hits)

Used for:

- communication networks
- distributed systems
- network design

Synchronizers [Awerbuch JACM '85]

Links with: Sparse Partition [Awerbuch et al. FOCS'90]; Distance Oracle [Thorup-Zwick STOC'01, Baswana et al. SODA'04]; Compact Routing [Peleg-Upfal STOC'89, Thorup-Zwick SPAA'01];

Variant: Geometric Spanners used for TSP (minimize $\sum_{e \in E(S)} \omega(e)$ of within a given stretch)

Outline

Extremal Graph Theory

(Bollobàs, Bondy, Erdös, Simonovits, ...)

We consider (unweighted) n-node graphs.

Theorem (Folklore – proved by Alon et al. '02)

A graph G without cycle of length $\leqslant 2k$ has $\leqslant \frac{1}{2}n^{1+1/k}$ edges.

Extremal Graph Theory

(Bollobàs, Bondy, Erdös, Simonovits, ...)

We consider (unweighted) n-node graphs.

Theorem (Folklore – proved by Alon et al. '02)

A graph G without cycle of length $\leqslant 2k$ has $\leqslant \frac{1}{2}n^{1+1/k}$ edges.

$\Leftrightarrow$ If $\mathrm{girth}(G) > 2k$, then $|E(G)| \leqslant \frac{1}{2}n^{1+1/k}$
(where $\mathrm{girth}(G)$ is the length of the smallest cycle of G).

Greedy Algorithm [Althöfer et al. '93]

Theorem

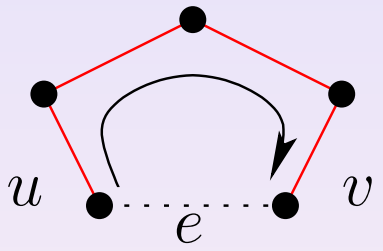
Every graph has a $(2k-1,0)$-spanner with $O(n^{1+1/k})$ edges.

Greedy Algorithm [Althöfer et al. '93]

Theorem

Every graph has a $(2k-1, 0)$-spanner with $O(n^{1+1/k})$ edges.

1. $S := \varnothing$ (the empty graph)
2. While $\exists e \in E(G)$ with stretch $> 2k-1$ in S, $S := S \cup \{e\}$



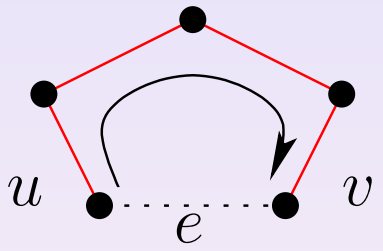
$k = 2$

Greedy Algorithm [Althöfer et al. '93]

Theorem

Every graph has a $(2k-1, 0)$-spanner with $O(n^{1+1/k})$ edges.

1. $S := \varnothing$ (the empty graph)
2. While $\exists e \in E(G)$ with stretch $> 2k-1$ in S, $S := S \cup \{e\}$



$k = 2$

Properties

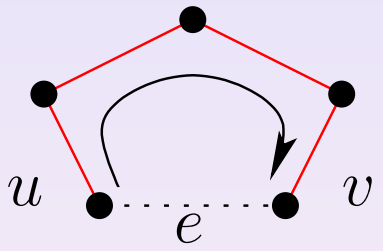
- $\text{stretch}(S) \leqslant 2k-1$

Greedy Algorithm [Althöfer et al. '93]

Theorem

Every graph has a $(2k-1, 0)$-spanner with $O(n^{1+1/k})$ edges.

1. $S := \varnothing$ (the empty graph)
2. While $\exists e \in E(G)$ with stretch $> 2k-1$ in S, $S := S \cup \{e\}$



$k = 2$

Properties

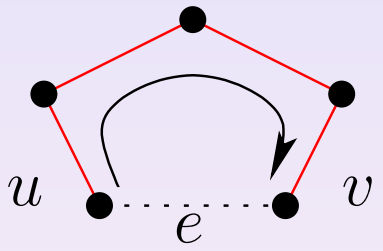
- $\text{stretch}(S) \leqslant 2k-1$
- Whenever $e = (u, v)$ is added to S, one cannot create any cycle of length $\leqslant 2k$

Greedy Algorithm [Althöfer et al. '93]

Theorem

Every graph has a $(2k-1, 0)$-spanner with $O(n^{1+1/k})$ edges.

1. $S := \varnothing$ (the empty graph)
2. While $\exists e \in E(G)$ with stretch $> 2k-1$ in S, $S := S \cup \{e\}$



$k = 2$

Properties

- $\text{stretch}(S) \leqslant 2k-1$
- Whenever $e = (u, v)$ is added to S, one cannot create any cycle of length $\leqslant 2k$

  Theorem [Folk] $\Rightarrow \text{size}(S) \leqslant \frac{1}{2} n^{1+1/k}$

Erdös-Simonovits Conjecture

Theorem [Folk] is optimal. More precisely,

Conjecture (ES82)

For each integer $k \geqslant 1$, there is a graph of girth $> 2k$ with at least $c_0 \cdot n^{1+1/k}$ edges, $0 < c_0 < \frac{1}{2}$.

Erdös-Simonovits Conjecture

Theorem [Folk] is optimal. More precisely,

Conjecture (ES82)

For each integer $k \geqslant 1$, there is a graph of girth $> 2k$ with at least $c_0 \cdot n^{1+1/k}$ edges, $0 < c_0 < \frac{1}{2}$.

[ES82] implies: $\exists$ (bipartite) graph B_k of girth $\geqslant 2k+2$ with $\frac{1}{2}c_0 \cdot n^{1+1/k}$ edges.

Erdös-Simonovits Conjecture

Theorem [Folk] is optimal. More precisely,

Conjecture (ES82)

For each integer $k \geqslant 1$, there is a graph of girth $> 2k$ with at least $c_0 \cdot n^{1+1/k}$ edges, $0 < c_0 < \frac{1}{2}$.

[ES82] implies: $\exists$ (bipartite) graph B_k of girth $\geqslant 2k+2$ with $\frac{1}{2}c_0 \cdot n^{1+1/k}$ edges.

$B_1 =$ complete bipartite.

Erdös-Simonovits Conjecture

Theorem [Folk] is optimal. More precisely,

Conjecture (ES82)

For each integer $k \geqslant 1$, there is a graph of girth $> 2k$ with at least $c_0 \cdot n^{1+1/k}$ edges, $0 < c_0 < \frac{1}{2}$.

[ES82] implies: $\exists$ (bipartite) graph B_k of girth $\geqslant 2k+2$ with $\frac{1}{2}c_0 \cdot n^{1+1/k}$ edges.

$B_1 =$ complete bipartite. [ES82] proved only for $k = 1, 2, 3, 5$.

For all k, $\exists$ graphs of girth $\geqslant 2k+2$ with $\Omega(n^{1+\frac{2}{3k}})$ edges.

Erdös-Simonovits Conjecture

For spanners [ES82] implies:

Corollary

If [ES82] holds, every (α, β)-spanner of B_k such that $\alpha + \beta < 2k + 1$ has size $|E(B_k)| = \Omega(n^{1+1/k})$.

Erdös-Simonovits Conjecture

For spanners [ES82] implies:

Corollary

If [ES82] holds, every (α, β)-spanner of B_k such that $\alpha + \beta < 2k + 1$ has size $|E(B_k)| = \Omega(n^{1+1/k})$.

(If we remove an edge (u, v) from B_k, then $\mathrm{stretch}(u, v) \geqslant 2k + 1$. But $\mathrm{stretch}(u, v) \leqslant \alpha \cdot d_G(u, v) + \beta = \alpha + \beta$, implies $\alpha + \beta \geqslant 2k + 1$: $\perp$)

Erdös-Simonovits Conjecture

For spanners [ES82] implies:

Corollary

If [ES82] holds, every (α, β)-spanner of B_k such that $\alpha + \beta < 2k + 1$ has size $|E(B_k)| = \Omega(n^{1+1/k})$.

(If we remove an edge (u, v) from B_k, then $\mathrm{stretch}(u, v) \geqslant 2k + 1$. But $\mathrm{stretch}(u, v) \leqslant \alpha \cdot d_G(u, v) + \beta = \alpha + \beta$, implies $\alpha + \beta \geqslant 2k + 1$: $\perp$)

Ex. for $k = 2$: a $(3, 0)$-spanner, a $(1, 2)$-spanner, or a $(4.98, 0.01)$-spanner, has $\Omega(n^{3/2})$ edges in the worst-case graph.

Recent Progress

[Woodruff FOCS'06]

For each $k \geqslant 1$, there is a graph such that every $(1, 2k-1)$-spanner requires $\Omega(\frac{1}{k} n^{1+1/k})$ edges.

So, "[ES82] is proved for $\alpha = 1$".

Outline

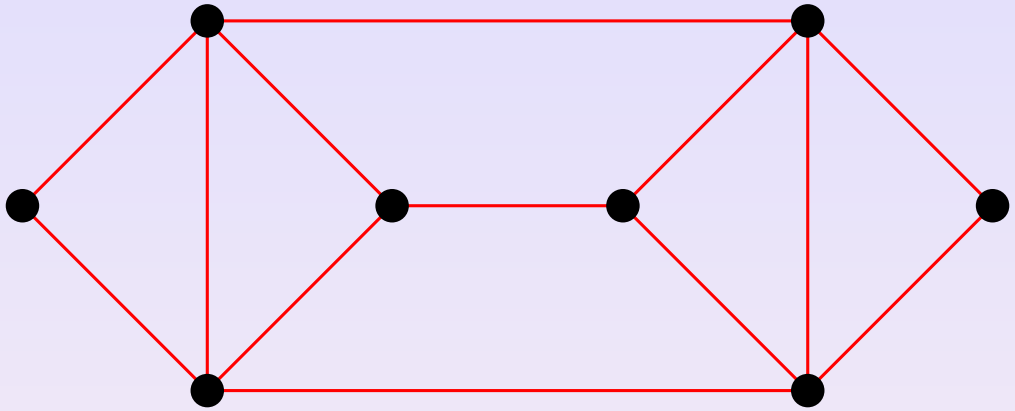
A $(3, 0)$-spanner of size $O(n^{3/2})$

(stretch < 3 implies $\Omega(n^2)$ edges, and stretch < 5 implies $\Omega(n^{3/2})$ edges)

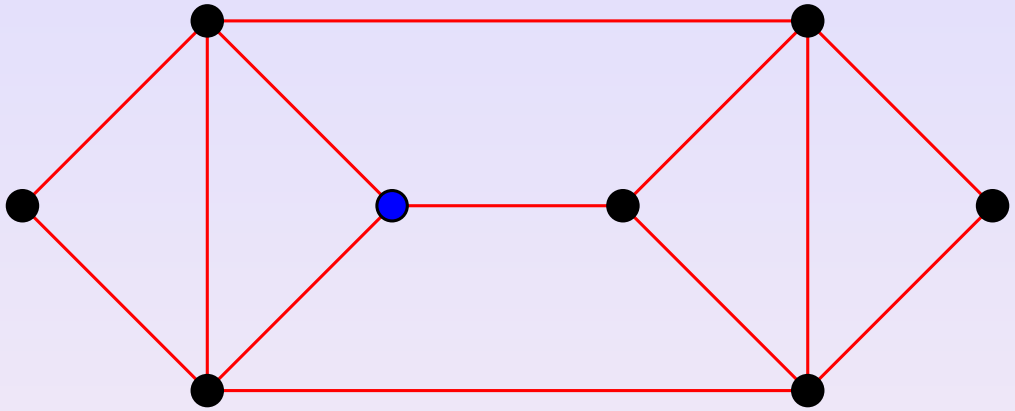
$B(u, r) =$ radius-r ball centered at u (in G)
$\mathrm{BFS}(u, X) =$ BFS tree rooted at u spanning X

1. $S := \varnothing$
2. While $\exists u \in V(G)$, $\deg(u) \geqslant \sqrt{n}$:
   1. $S := S \cup \mathrm{BFS}(u, B(u, 2))$
   2. $G := G \setminus B(u, 1)$
3. $S := S \cup G$

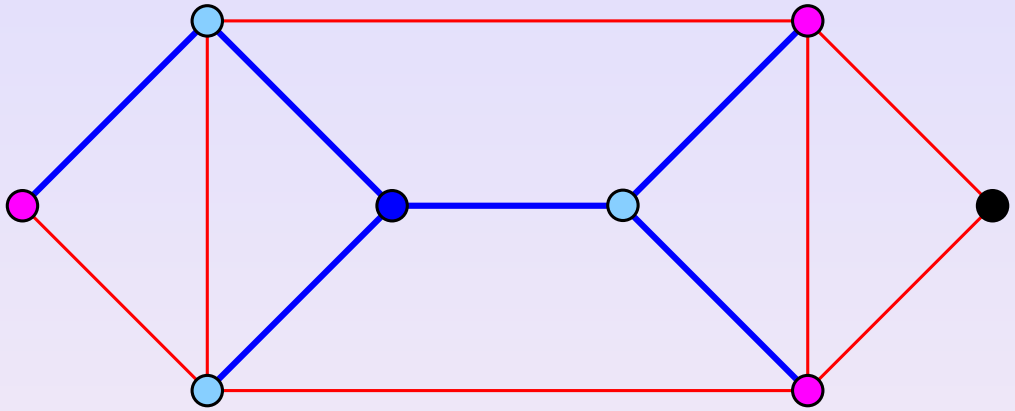
Example: $n = 8$, $\sqrt{n} \approx 2.82$



Example: $n = 8$, $\sqrt{n} \approx 2.82$



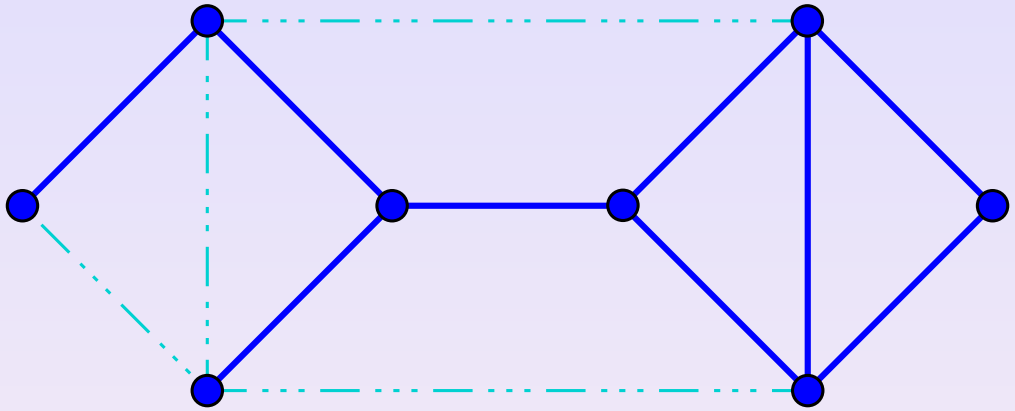
Example: $n = 8$, $\sqrt{n} \approx 2.82$



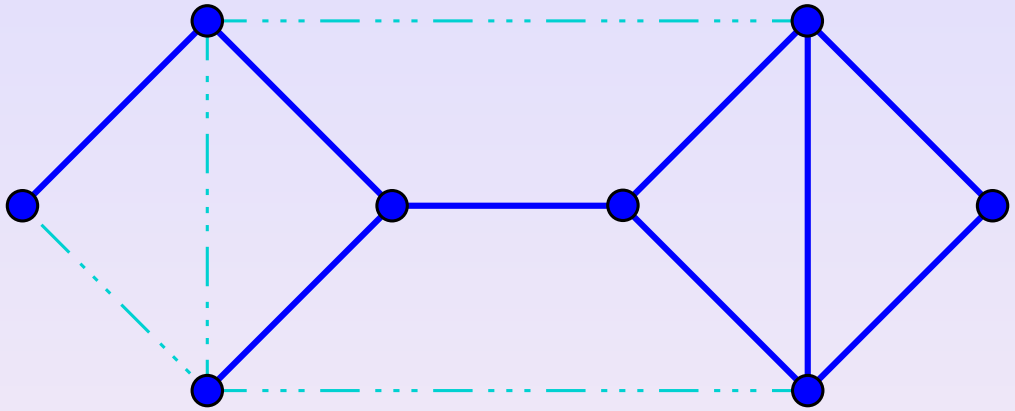
Example: $n = 8$, $\sqrt{n} \approx 2.82$

Example: $n = 8$, $\sqrt{n} \approx 2.82$

Example: $n = 8$, $\sqrt{n} \approx 2.82$

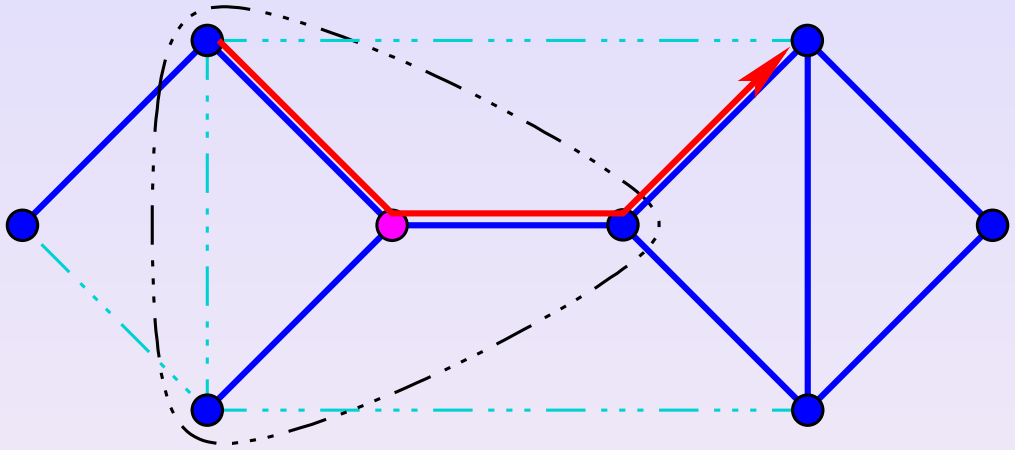


Example: $n = 8$, $\sqrt{n} \approx 2.82$



Size: $\sqrt{n} \times (n-1) + n \times \sqrt{n} < 2n\sqrt{n}$

Example: $n = 8$, $\sqrt{n} \approx 2.82$



Size: $\sqrt{n} \times (n-1) + n \times \sqrt{n} < 2n\sqrt{n}$
Stretch: 3

A $(1, 2)$-spanners of size $O(n^{3/2})$

(Aingworth et al. '99)

1. $S := \varnothing$
2. While $\exists u \in V(G)$, $\deg(u) \geqslant \sqrt{n}$:
   1. $S := S \cup \mathrm{BFS}(u, G)$
   2. $G := G \setminus B(u, 1)$
3. $S := S \cup G$

A $(1,2)$-spanners of size $O(n^{3/2})$

(Aingworth et al. '99)

1. $S := \varnothing$
2. While $\exists u \in V(G)$, $\deg(u) \geqslant \sqrt{n}$:
   1. $S := S \cup \mathrm{BFS}(u, G)$
   2. $G := G \setminus B(u, 1)$
3. $S := S \cup G$

Size: $\leqslant 2n\sqrt{n}$

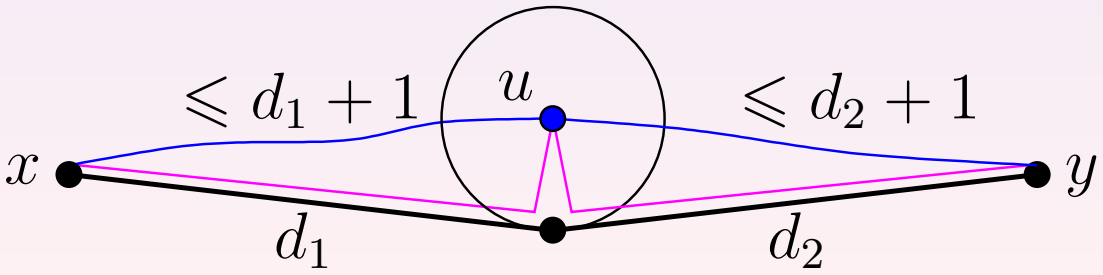
A $(1,2)$-spanners of size $O(n^{3/2})$

(Aingworth et al. '99)

1. $S := \varnothing$
2. While $\exists u \in V(G)$, $\deg(u) \geqslant \sqrt{n}$:
   1. $S := S \cup \mathrm{BFS}(u, G)$
   2. $G := G \setminus B(u, 1)$
3. $S := S \cup G$

Size: $\leqslant 2n\sqrt{n}$

Stretch: if an xy-shortest path does not intersect a neighbor of any selected node u (or cuts u), then stretch is 1, otherwise $d_S(x, y) \leqslant d_1 + d_2 + 2 = d_G(x, y) + 2$.



Other Results

[Baswana, Pettie et al. SODA '05]

Every graph has a $(1, 6)$-spanner of size $O(n^{4/3})$, and a $(k, k-1)$-spanner of size $O(kn^{1+1/k})$.

Non-trivial construction & analysis. If the Woodruff's lower bound is tight, stretch of $(1, 4)$ for a size $O(n^{4/3})$ is possible.

Other Results

[Baswana, Pettie et al. SODA '05]

Every graph has a $(1, 6)$-spanner of size $O(n^{4/3})$, and a $(k, k-1)$-spanner of size $O(kn^{1+1/k})$.

Non-trivial construction & analysis. If the Woodruff's lower bound is tight, stretch of $(1, 4)$ for a size $O(n^{4/3})$ is possible.

Open questions

Does exist for every graph:

- a $(1, \beta)$-spanner of size $o(n^{4/3})$ for some constant β?
- a $(1, f(k))$-spanner of size $O(n^{1+1/k})$ for some f?

Outline

The Model

LOCAL model: (a.k.a. Free model, or Linial's model)

- synchrone
- unique IDs
- no size limit messages
- no failures
- simultaneous wake-up
- arbitrary computational power at nodes

Time complexity: number of rounds
(1 round = messages sent/received between all neighbors)

Idea #1: distributed greedy algorithm

For every node u do:

1. $S_u := \varnothing$ and $A_u := \{\mathrm{id}(u)\}$ [u is active]
2. At each round add edge (u, v) to S_u if:
   - $\mathrm{id}(u)$ is minimum in $\bigcup_{w \in B(u,k)} A_w$; and
   - (u, v) does not create a cycle of length $\leqslant 2k$ in the current graph $\bigcup_{w \in B(u,k)} S_w$.
3. Repeat 2 until all incident edges u have been tested.
4. $A_u := \varnothing$ [u becomes non active]

Idea #1: distributed greedy algorithm

For every node u do:

1. $S_u := \varnothing$ and $A_u := \{\mathrm{id}(u)\}$ [u is active]
2. At each round add edge (u, v) to S_u if:
   - $\mathrm{id}(u)$ is minimum in $\bigcup_{w \in B(u,k)} A_w$; and
   - (u, v) does not create a cycle of length $\leqslant 2k$ in the current graph $\bigcup_{w \in B(u,k)} S_w$.
3. Repeat 2 until all incident edges u have been tested.
4. $A_u := \varnothing$ [u becomes non active]

Size: $O(n^{1+1/k})$ since no cycles of length $\leqslant 2k$

Idea #1: distributed greedy algorithm

For every node u do:

1. $S_u := \varnothing$ and $A_u := \{\mathrm{id}(u)\}$ [u is active]
2. At each round add edge (u, v) to S_u if:
   - $\mathrm{id}(u)$ is minimum in $\bigcup_{w \in B(u,k)} A_w$; and
   - (u, v) does not create a cycle of length $\leqslant 2k$ in the current graph $\bigcup_{w \in B(u,k)} S_w$.
3. Repeat 2 until all incident edges u have been tested.
4. $A_u := \varnothing$ [u becomes non active]

Size: $O(n^{1+1/k})$ since no cycles of length $\leqslant 2k$
Stretch: $2k - 1$

Idea #1: distributed greedy algorithm

For every node u do:

1. $S_u := \varnothing$ and $A_u := \{\mathrm{id}(u)\}$ [u is active]
2. At each round add edge (u, v) to S_u if:
   - $\mathrm{id}(u)$ is minimum in $\bigcup_{w \in B(u,k)} A_w$; and
   - (u, v) does not create a cycle of length $\leqslant 2k$ in the current graph $\bigcup_{w \in B(u,k)} S_w$.
3. Repeat 2 until all incident edges u have been tested.
4. $A_u := \varnothing$ [u becomes non active]

Size: $O(n^{1+1/k})$ since no cycles of length $\leqslant 2k$
Stretch: $2k - 1$
Time: $O(|E(G)|) = O(n^2)$ rounds !!!

Idea #2: alternate distributed greedy (for $k = 2$)

(Derbel et al. 2006)

For every node u do:

1. $S_u := \varnothing$
2. Compute its degree d_u in G
3. if $d_u \geqslant \sqrt{n}$ and $\mathrm{id}(u)$ is minimum in $\bigcup_{w \in B(u,2), d_w \geqslant \sqrt{n}} \mathrm{id}(w)$, then do:
   - $S_u := S_u \cup \mathrm{BFS}(u, B(u,2))$
   - Broadcast S_u in $B(u,2)$
   - $G := G \setminus B(u,1)$
4. Repeat 2-3 $\sqrt{n}$ times
5. $S_u := S_u \cup B(u,1)$

Idea #2: alternate distributed greedy (for $k = 2$)

(Derbel et al. 2006)

For every node u do:

1. $S_u := \varnothing$
2. Compute its degree d_u in G
3. if $d_u \geqslant \sqrt{n}$ and $\mathrm{id}(u)$ is minimum in $\bigcup_{w \in B(u,2), d_w \geqslant \sqrt{n}} \mathrm{id}(w)$, then do:
   - $S_u := S_u \cup \mathrm{BFS}(u, B(u,2))$
   - Broadcast S_u in $B(u,2)$
   - $G := G \setminus B(u,1)$
4. Repeat 2-3 $\sqrt{n}$ times
5. $S_u := S_u \cup B(u,1)$

Stretch: 3
Size: $\leqslant 2n\sqrt{n}$
Time: $O(\sqrt{n})$ rounds [and $O(n^{1-1/k})$]

Idea #3: fast greedy

(Panconesi et al. 2005)

For every node u do:

1. $S_u := B(u, 1)$
2. Remove edge (u, v) from S_u if $(u, v) \in$ cycle C of length $\leqslant 2k$ and if $\langle \mathrm{id}(u), \mathrm{id}(v) \rangle$ is minimum on C.

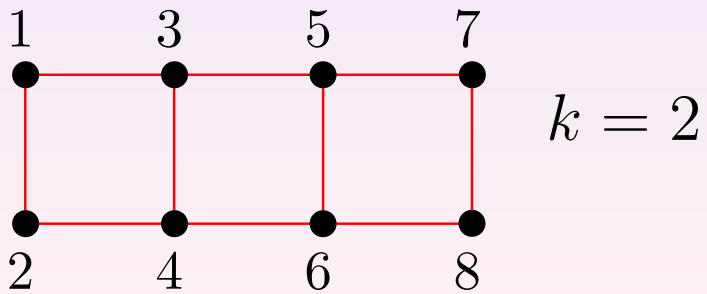
Idea #3: fast greedy

(Panconesi et al. 2005)

For every node u do:

1. $S_u := B(u, 1)$
2. Remove edge (u, v) from S_u if $(u, v) \in$ cycle C of length $\leqslant 2k$ and if $\langle \mathrm{id}(u), \mathrm{id}(v) \rangle$ is minimum on C.

Time: k rounds
Size: $O(n^{1+1/k})$ since no cycle of length $\leqslant 2k$
Stretch:



$k = 2$

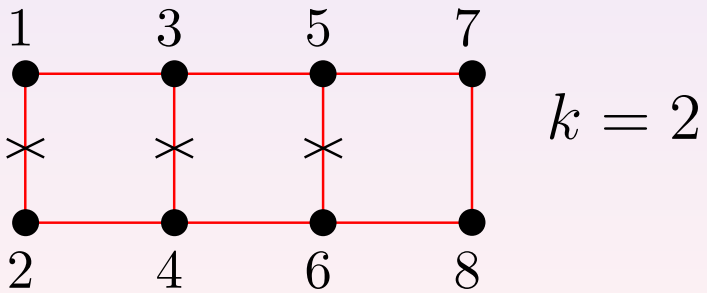
Idea #3: fast greedy

(Panconesi et al. 2005)

For every node u do:

1. $S_u := B(u, 1)$
2. Remove edge (u, v) from S_u if $(u, v) \in$ cycle C of length $\leqslant 2k$ and if $\langle \mathrm{id}(u), \mathrm{id}(v) \rangle$ is minimum on C.

Time: k rounds
Size: $O(n^{1+1/k})$ since no cycle of length $\leqslant 2k$
Stretch: unbounded ! (however G remains connected)



$k = 2$

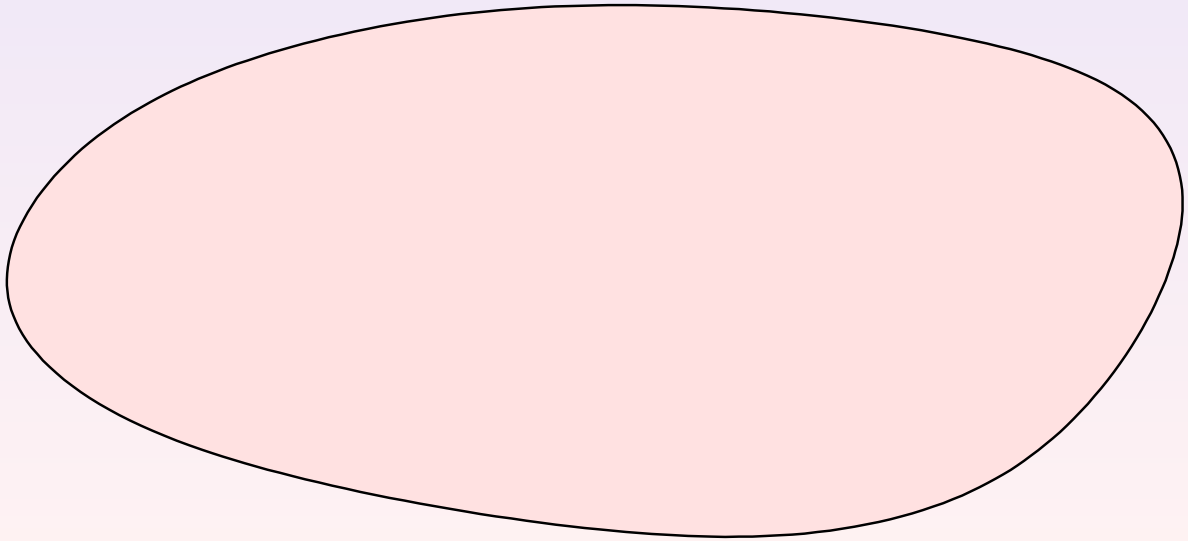
Conclusion for ideas #1,#2, and #3

Distributed versions of greedy algorithms are **non efficient**, or does not garantee stretch-size tradeoff. To do faster, one need parallelism! Greedy algorithms are inherently **non-parallel**.

Idea #4: fast clustering algorithm

(Awerbuch-Peleg ... in the 90')

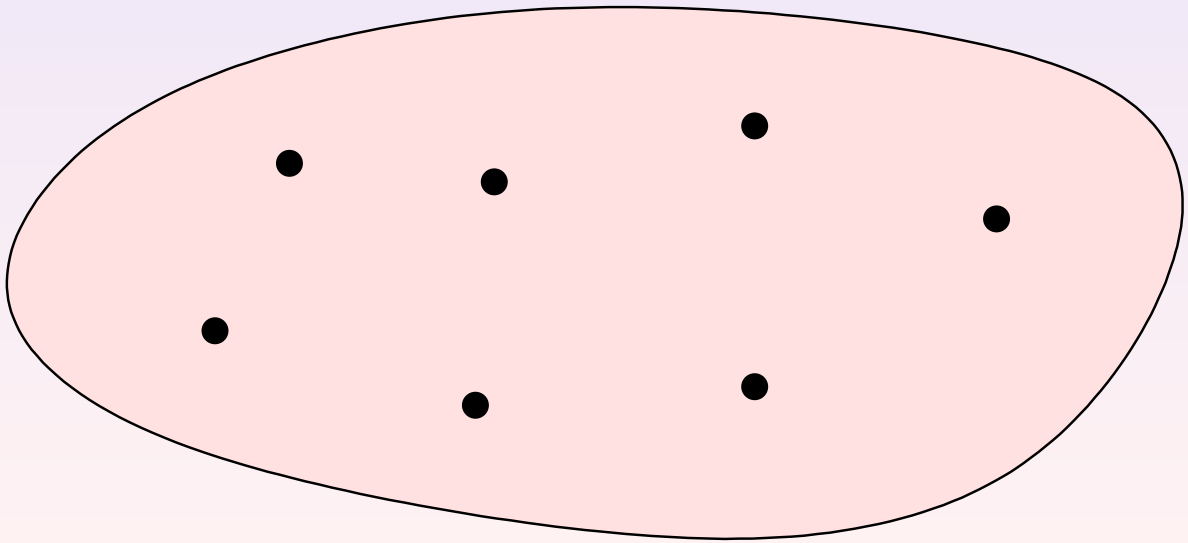
1. Compute an maximal indendent set (MIS) X in G^2



Idea #4: fast clustering algorithm

(Awerbuch-Peleg ... in the 90')

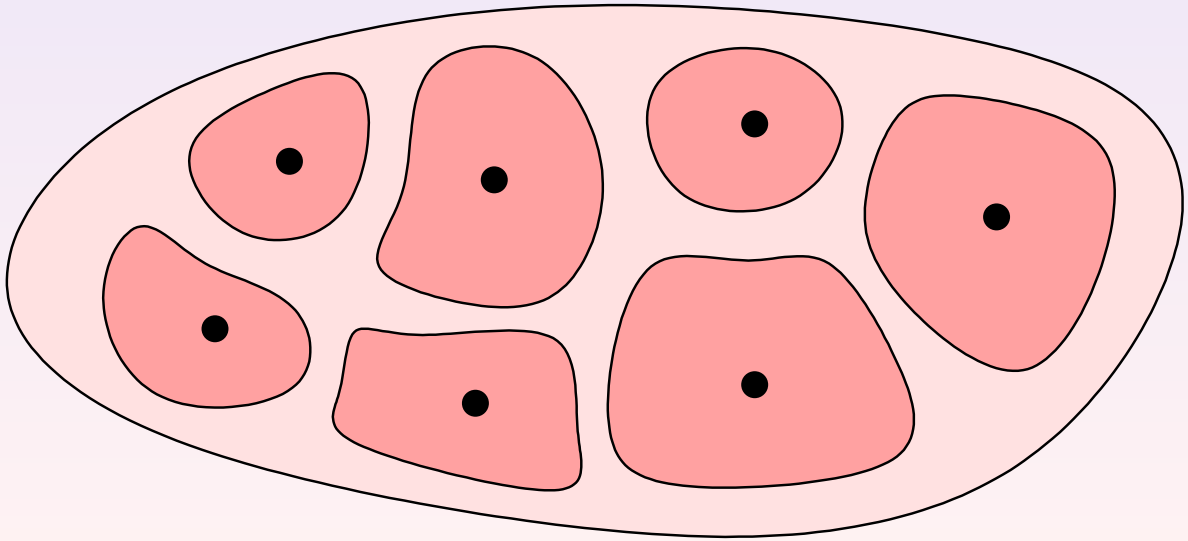
1. Compute an maximal indendent set (MIS) X in G^2
   $\Rightarrow$ points pariwise at distance $\geqslant 3$ and $\leqslant 5$



Idea #4: fast clustering algorithm

(Awerbuch-Peleg ... in the 90')

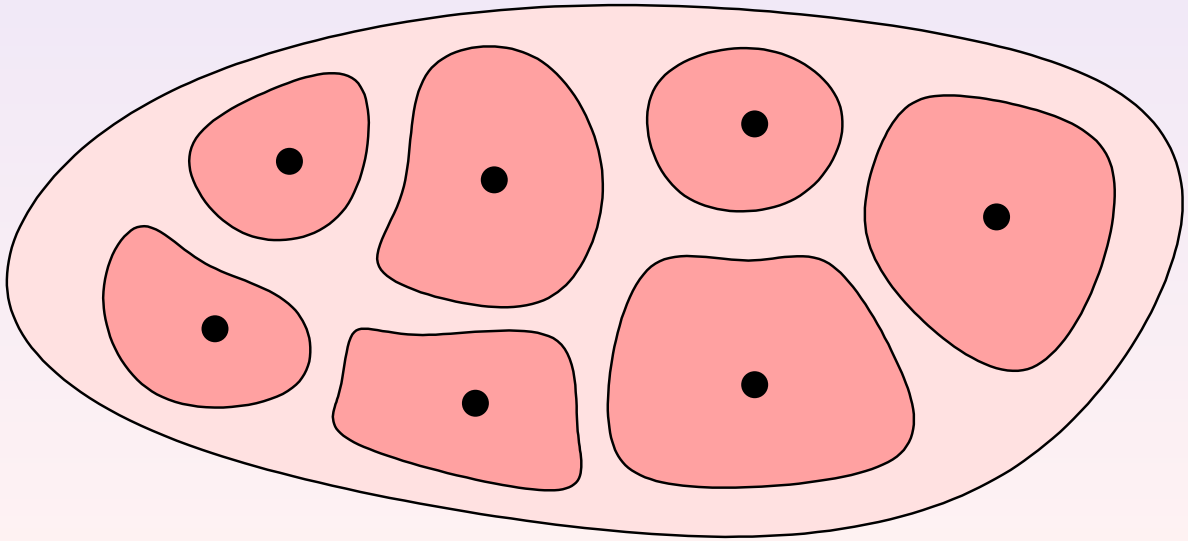
1. Compute an maximal indendent set (MIS) X in G^2
   $\Rightarrow$ points pariwise at distance $\geqslant 3$ and $\leqslant 5$
2. Create (vote!) independent regions with all centers in X



Idea #4: fast clustering algorithm

(Awerbuch-Peleg ... in the 90')

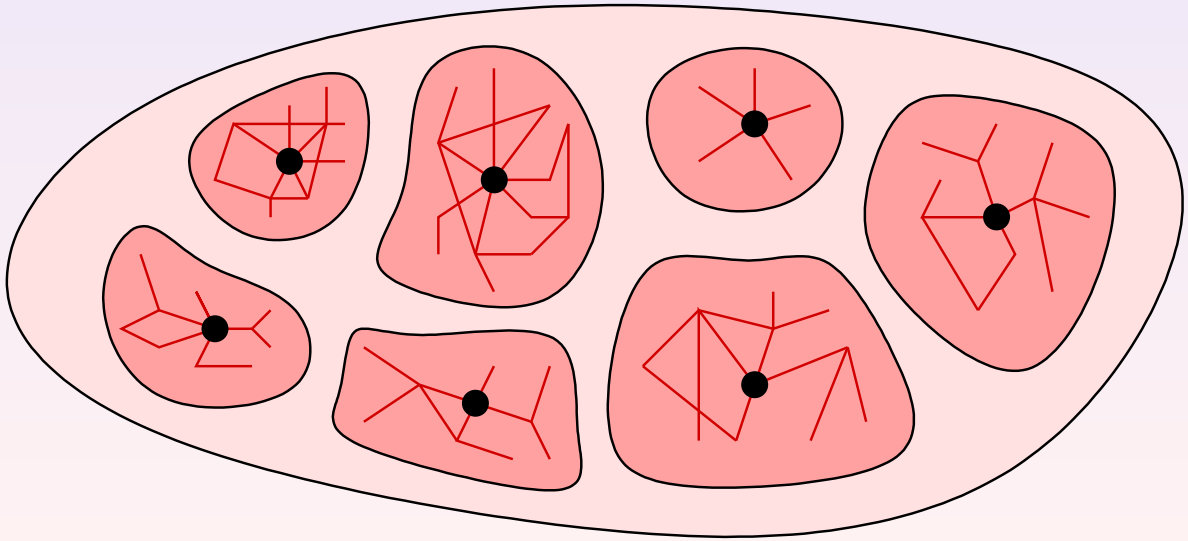
1. Compute an maximal indendent set (MIS) X in G^2
   $\Rightarrow$ points pariwise at distance $\geqslant 3$ and $\leqslant 5$
2. Create (vote!) independent regions with all centers in X
   $\Rightarrow$ regions of radius $\leqslant 2$



Idea #4: fast clustering algorithm

(Awerbuch-Peleg ... in the 90')

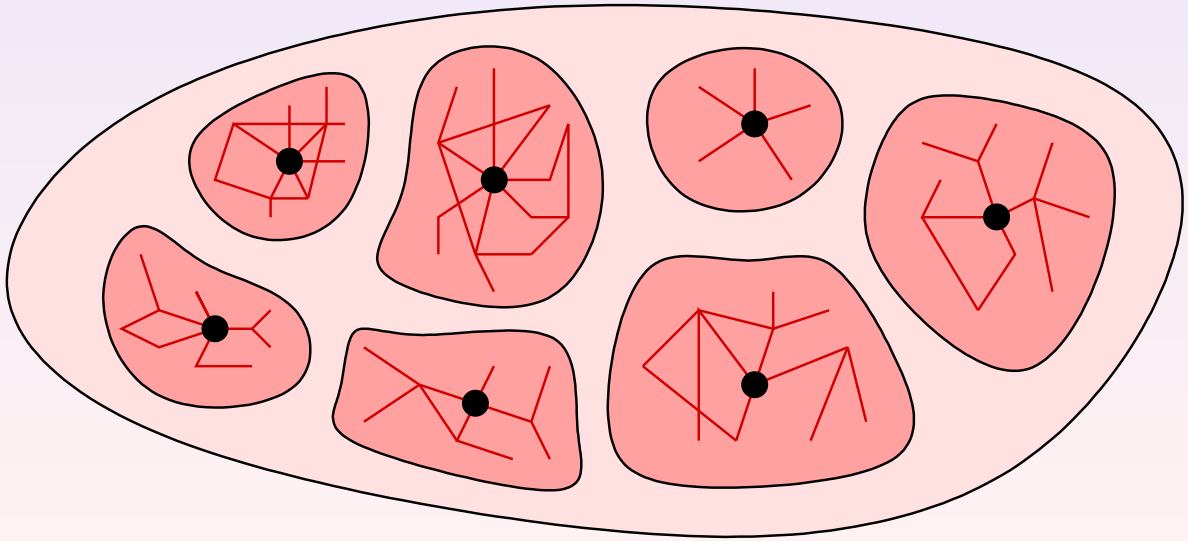
1. Compute an maximal indendent set (MIS) X in G^2
   $\Rightarrow$ points pariwise at distance $\geqslant 3$ and $\leqslant 5$
2. Create (vote!) independent regions with all centers in X
   $\Rightarrow$ regions of radius $\leqslant 2$
3. In parallel compute a "good" spanner in each region



Idea #4: fast clustering algorithm

(Awerbuch-Peleg ... in the 90')

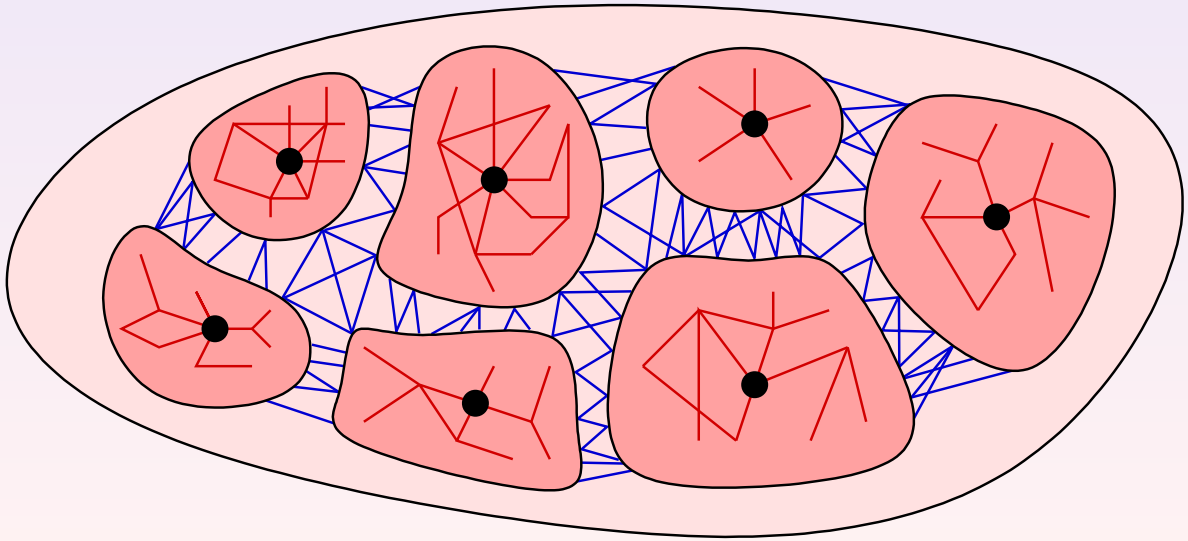
1. Compute an maximal indendent set (MIS) X in G^2
   $\Rightarrow$ points pariwise at distance $\geqslant 3$ and $\leqslant 5$
2. Create (vote!) independent regions with all centers in X
   $\Rightarrow$ regions of radius $\leqslant 2$
3. In parallel compute a "good" spanner in each region
   $\Rightarrow$ optimal stretch-size tradeoff in $O(1)$ rounds



Idea #4: fast clustering algorithm

(Awerbuch-Peleg ... in the 90')

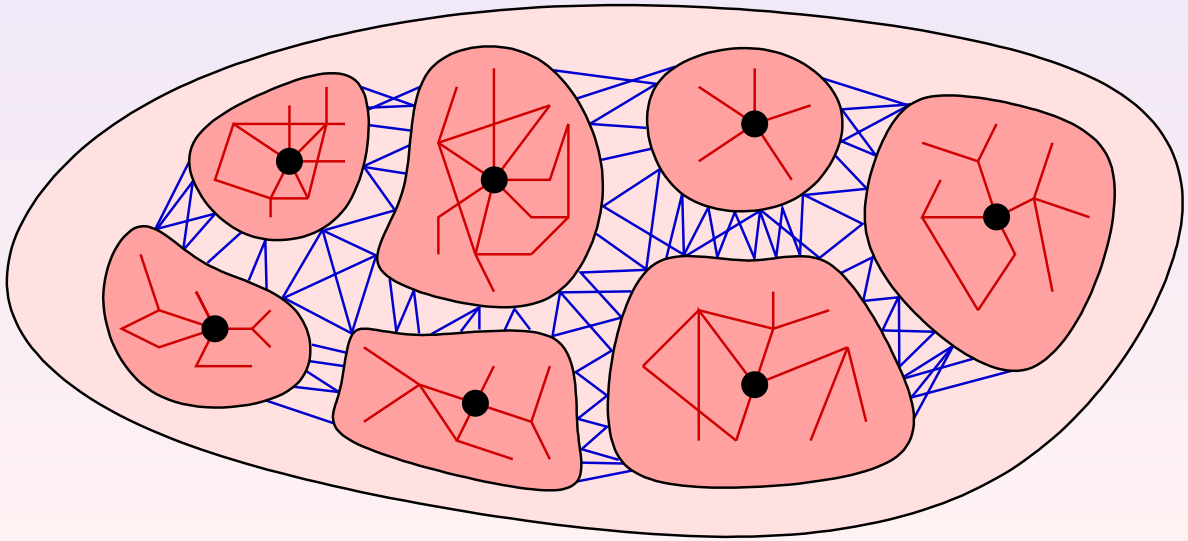
1. Compute an maximal indendent set (MIS) X in G^2
   $\Rightarrow$ points pariwise at distance $\geqslant 3$ and $\leqslant 5$
2. Create (vote!) independent regions with all centers in X
   $\Rightarrow$ regions of radius $\leqslant 2$
3. In parallel compute a "good" spanner in each region
   $\Rightarrow$ optimal stretch-size tradeoff in $O(1)$ rounds
4. Cover inter-region edges (bipartite) with length-3 paths



Idea #4: fast clustering algorithm

(Awerbuch-Peleg ... in the 90')

1. Compute an maximal indendent set (MIS) X in G^2
   $\Rightarrow$ points pariwise at distance $\geqslant 3$ and $\leqslant 5$
2. Create (vote!) independent regions with all centers in X
   $\Rightarrow$ regions of radius $\leqslant 2$
3. In parallel compute a "good" spanner in each region
   $\Rightarrow$ optimal stretch-size tradeoff in $O(1)$ rounds
4. Cover inter-region edges (bipartite) with length-3 paths
   doable with stretch 3 and size $n + |R_u|\sqrt{n}$ for R_u



Idea #4: analysis

Size: $O(n^{3/2})$ [and $O(kn^{1+1/k})$]

Stretch: 3 [and $4k - O(1)$]

Idea #4: analysis

Size: $O(n^{3/2})$ [and $O(kn^{1+1/k})$]
Stretch: 3 [and $4k - O(1)$]
Time: $O(\mathrm{MIS}(n))$ [and $O(f(k) \cdot \mathrm{MIS}(n))$]

Time complexity for distributed MIS (widly open):

- Upper bound: $2^{O(\sqrt{\log n})}$ [Panconesi et al. STOC'96]
- Lower bound: $\Omega(\sqrt{\log n / \log\log n})$ [Kuhn et al. PODC'06]

In sequential, MIS is computed by a purely greedy algorithm ...

1. $X := \varnothing$
2. while $\exists u \in V(G)$, $X := X \cup \{u\}$ and $G := G \setminus B(u, 1)$

Idea #4: refinement

It is possible to avoid the MIS computation, and to compute **an independent $O(\log n)$-dominating set**, which can be done in $O(\log n)$ time.

[Derbel et al. DISC'07]

Time: $2^{O(k)} \log^{k-1} n$ rounds
Size: $O(kn^{1+1/k})$
Stretch: $4k - 5$

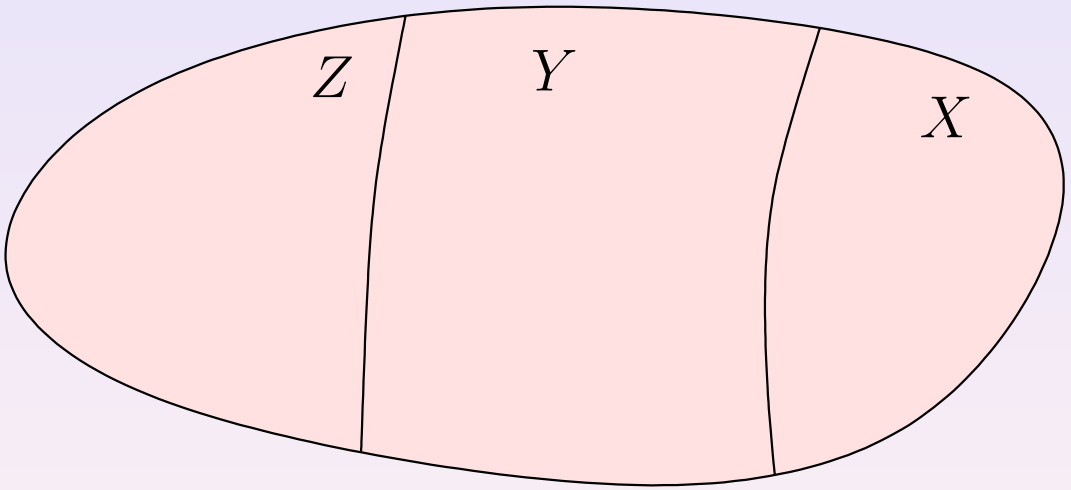
For $k = 2$:
$\Rightarrow$ a stretch-3 spanner of size $O(n^{3/2})$ in $O(\log n)$ time

Can we do faster?

Idea #5: randomization! whenever there is no more ideas

(Baswana et al. '05)

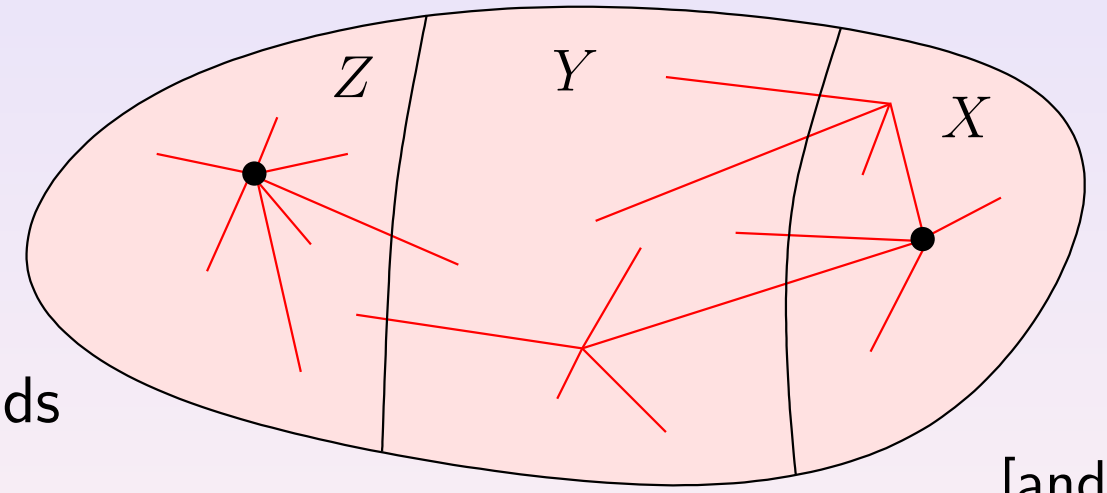
1. $x_u := 1|0$ with proba $1/\sqrt{n}$. Let $X := \{u \mid x_u = 1\}$
2. if $B(u,1) \cap X = \varnothing$, then $S_u := B(u,1)$
3. if $x_u = 1$, then $S_u := \mathrm{BFS}(u, B(u,2))$



Idea #5: randomization! whenever there is no more ideas

(Baswana et al. '05)

1. $x_u := 1|0$ with proba $1/\sqrt{n}$. Let $X := \{u \mid x_u = 1\}$
2. if $B(u,1) \cap X = \varnothing$, then $S_u := B(u,1)$
3. if $x_u = 1$, then $S_u := \mathrm{BFS}(u, B(u,2))$



Time: 2 rounds [and k]

Stretch: 3 [and $2k-1$]

Size: $2n^{3/2}$ in expectation [and $O(kn^{1+1/k})$]

(In expectation: $|X| = \sqrt{n}$, and if $u \in Z$, then $\deg(u) \leqslant \sqrt{n}$)

Outline

The Question

What is the smallest t such that if each node u of a graph knows $B(u,t)$, then u can **deterministically** decide alone which incident edges to keep to form a $(3,0)$-spanner of size $O(n^{3/2})$?

The Question

What is the smallest t such that if each node u of a graph knows $B(u,t)$, then u can **deterministically** decide alone which incident edges to keep to form a $(3,0)$-spanner of size $O(n^{3/2})$?

Theorem (Derbel, G., Peleg, Viennot - PODC'08)

There is a determinist distributed algorithm that for every n-node graph computes a $(2k-1,0)$-spanner of size at most $kn^{1+1/k}$ in time k.

Moreover, if n is unkown, then the algorithm requires time $2k-1$. Also k is the best possible time bound, even "expected time" for a randomized (Las Vegas) algorithm.

Second result

Theorem (2)

For every $\varepsilon \in (0, 2]$, there is a determinist distributed algorithm that for every n-node graph (where n is unkown to the nodes) computes a $(1+\varepsilon, 2)$-spanner of size $O(\varepsilon^{-1} n^{3/2})$ in $O(\varepsilon^{-1})$ time.

We can also show that $(1+\varepsilon, 2)$-spanner of size $O(n^{3/2})$ cannot be computed in less than $\Omega(\varepsilon^{-1})$ expected time.

The Algorithm (for $k = 2$)

For every node u do:

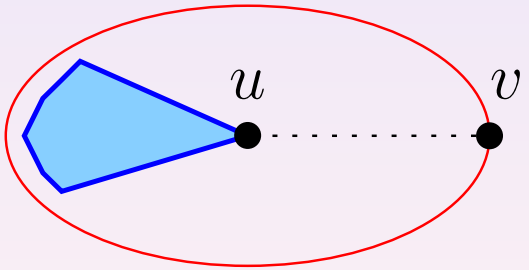
1. $R_u := \{u\} \cup \{$ **any** selection of $\lfloor \sqrt{|B(u,3)|} \rfloor$ neighbors $\}$
2. send R_u and receive R_v to/from its neighbors
3. $W := B(u,1) \setminus \{v \mid u \in R_u\} \setminus R_u$ and $C_u := \varnothing$
4. while $\exists w \in W$:
   1. $C_u := C_u \cup \{w\}$
   2. $W := W \setminus \{v \in W \mid R_v \cap R_w \neq \varnothing\}$
5. Add edges from u to $R_u \cup C_u$

The Algorithm (for $k = 2$)

For every node u do:

1. $R_u := \{u\} \cup \{$ **any** selection of $\lfloor \sqrt{|B(u,3)|} \rfloor$ neighbors $\}$
2. send R_u and receive R_v to/from its neighbors
3. $W := B(u,1) \setminus \{v \mid u \in R_u\} \setminus R_u$ and $C_u := \varnothing$
4. while $\exists w \in W$:
   1. $C_u := C_u \cup \{w\}$
   2. $W := W \setminus \{v \in W \mid R_v \cap R_w \neq \varnothing\}$
5. Add edges from u to $R_u \cup C_u$

Stretch: if $(u,v) \notin S$, then v removed from W in 4.2.

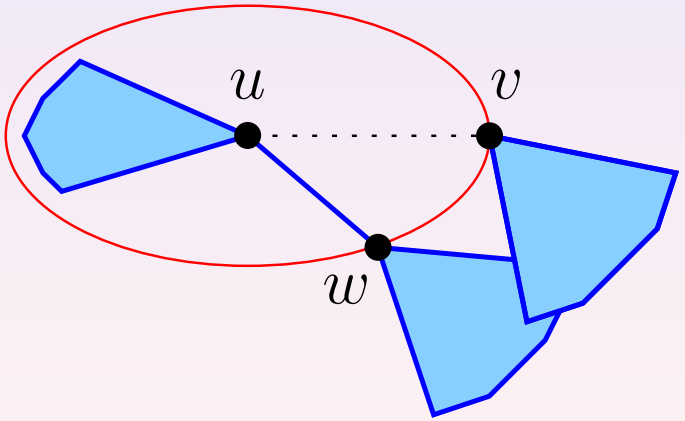


The Algorithm (for $k = 2$)

For every node u do:

1. $R_u := \{u\} \cup \{$ **any** selection of $\lfloor \sqrt{|B(u,3)|} \rfloor$ neighbors $\}$
2. send R_u and receive R_v to/from its neighbors
3. $W := B(u,1) \setminus \{v \mid u \in R_u\} \setminus R_u$ and $C_u := \varnothing$
4. while $\exists w \in W$:
   1. $C_u := C_u \cup \{w\}$
   2. $W := W \setminus \{v \in W \mid R_v \cap R_w \neq \varnothing\}$
5. Add edges from u to $R_u \cup C_u$

Stretch: if $(u,v) \notin S$, then v removed from W in 4.2. Thus $\exists w \in C_u$ with $R_v \cap R_w \neq \varnothing$. Hence, stretch 3.

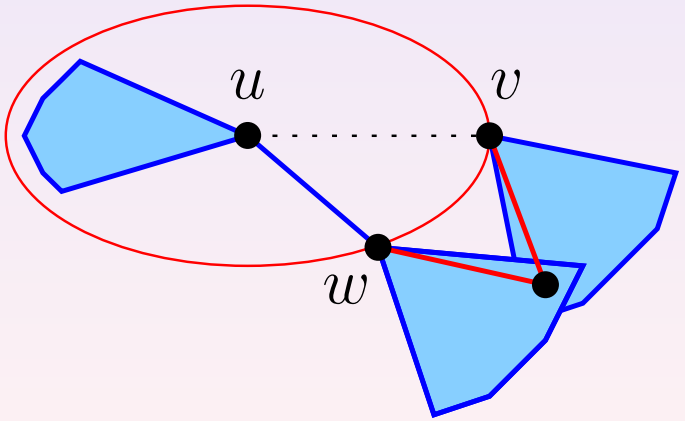


The Algorithm (for $k = 2$)

For every node u do:

1. $R_u := \{u\} \cup \{$ **any** selection of $\lfloor \sqrt{|B(u,3)|} \rfloor$ neighbors $\}$
2. send R_u and receive R_v to/from its neighbors
3. $W := B(u,1) \setminus \{v \mid u \in R_u\} \setminus R_u$ and $C_u := \varnothing$
4. while $\exists w \in W$:
   1. $C_u := C_u \cup \{w\}$
   2. $W := W \setminus \{v \in W \mid R_v \cap R_w \neq \varnothing\}$
5. Add edges from u to $R_u \cup C_u$

Stretch: if $(u,v) \notin S$, then v removed from W in 4.2. Thus $\exists w \in C_u$ with $R_v \cap R_w \neq \varnothing$. Hence, stretch 3.

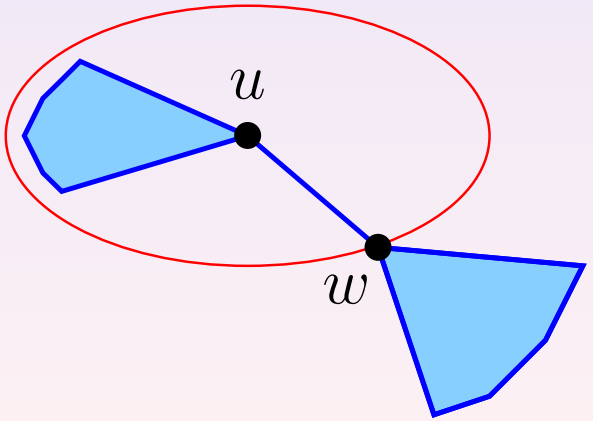


The Algorithm (for $k = 2$)

For every node u do:

1. $R_u := \{u\} \cup \{$ **any** selection of $\lfloor \sqrt{|B(u,3)|} \rfloor$ neighbors $\}$
2. send R_u and receive R_v to/from its neighbors
3. $W := B(u,1) \setminus \{v \mid u \in R_u\} \setminus R_u$ and $C_u := \varnothing$
4. while $\exists w \in W$:
   1. $C_u := C_u \cup \{w\}$
   2. $W := W \setminus \{v \in W \mid R_v \cap R_w \neq \varnothing\}$
5. Add edges from u to $R_u \cup C_u$

Size: $w \in W$ after 3 has degree $\geqslant |B(w,3)|^{1/2}$. Thus, $\deg(w) \geqslant |B(u,2)|^{1/2}$ since $B(u,2) \subseteq B(w,3)$.

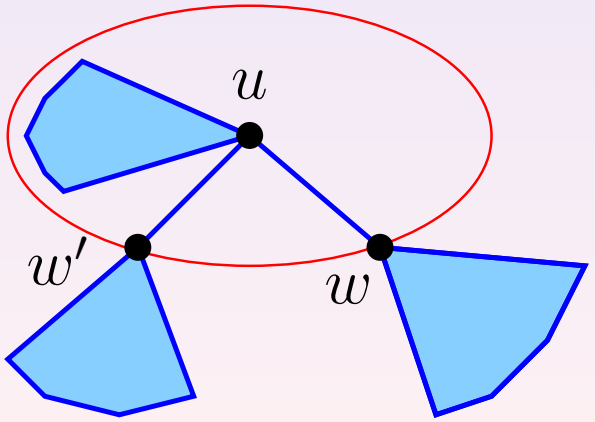


The Algorithm (for $k = 2$)

For every node u do:

1. $R_u := \{u\} \cup \{$ **any** selection of $\lfloor \sqrt{|B(u,3)|} \rfloor$ neighbors $\}$
2. send R_u and receive R_v to/from its neighbors
3. $W := B(u,1) \setminus \{v \mid u \in R_u\} \setminus R_u$ and $C_u := \varnothing$
4. while $\exists w \in W$:
   1. $C_u := C_u \cup \{w\}$
   2. $W := W \setminus \{v \in W \mid R_v \cap R_w \neq \varnothing\}$
5. Add edges from u to $R_u \cup C_u$

Size: $w \in W$ after 3 has degree $\geqslant |B(w,3)|^{1/2}$. Thus, $\deg(w) \geqslant |B(u,2)|^{1/2}$ since $B(u,2) \subseteq B(w,3)$. R_w taken in C_u are disjoint. #loops is $|C_u| \leqslant |B(u,2)|/|B(u,2)|^{1/2} \leqslant \sqrt{n}$. Size: $n \times 2\sqrt{n}$.



Outline

Conclusion

- The computation of "optimal" sparse spanners is LOCAL
- How to reduce message size? (lower bound?)
- Does exist $(1, \beta)$-spanner of size $o(n^{4/3})$?

Thank You
for your attention