

Routing in Distributed Systems

- Lecture 2 -

Cyril Gavoille

University of Bordeaux, France

April 6-9, 2009
Valparaíso, Chile



- ① Routing with Succinct Tables
- ② Constructing Sparse Spanners
- ③ Routing v.s. Spanners

- Lecture 2 -

Constructing Sparse Spanners

Outline

Introduction, Examples & Definitions

On the Combinatorial Problem

Tree Cover Algorithms for $k = 2$

A Distributed Algorithm

Conclusion

Outline

Introduction, Examples & Definitions

On the Combinatorial Problem

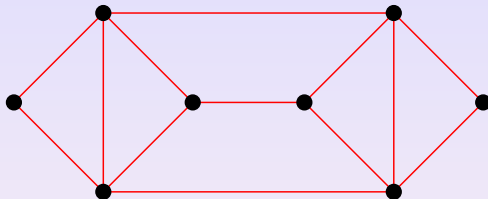
Tree Cover Algorithms for $k = 2$

A Distributed Algorithm

Conclusion

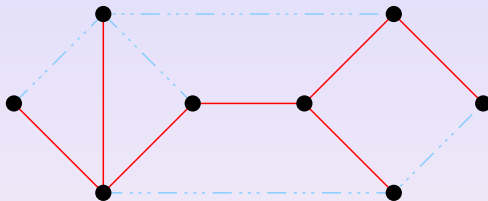
What is a Spanner?

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



What is a Spanner?

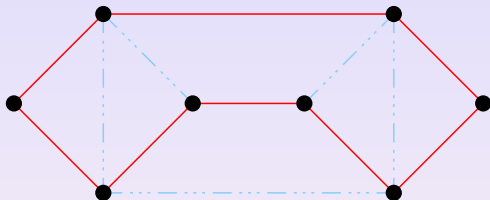
A **spanner** of a graph G is a subgraph spanning $V(G)$



- a spanning tree

What is a Spanner?

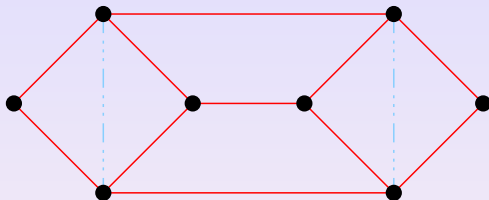
A **spanner** of a graph G is a subgraph spanning $V(G)$



- 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 .

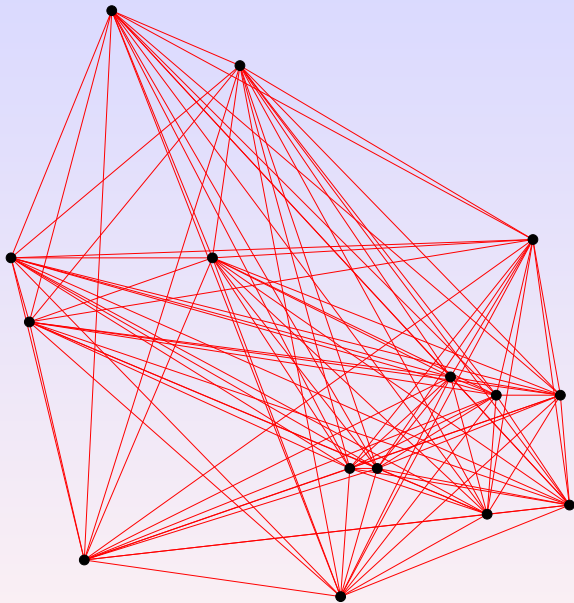
Approximate Distance Spanners

There are two “natural” criteria for a spanner of G :

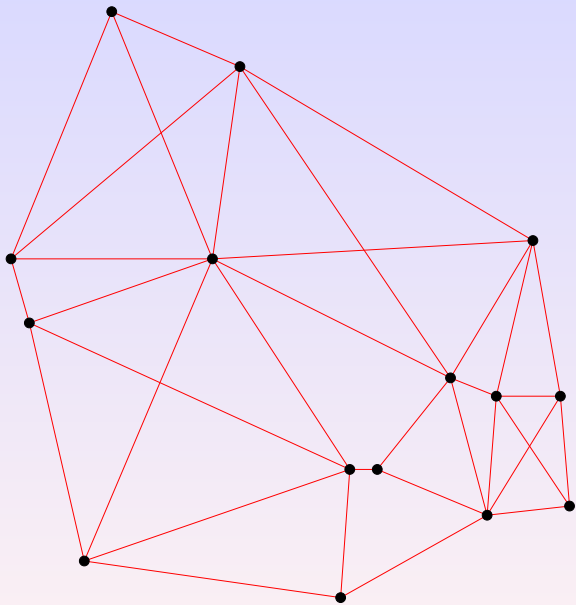
- **size:** its number of edges.
- **stretch:** its maximum distance distortion from G .

Goals:

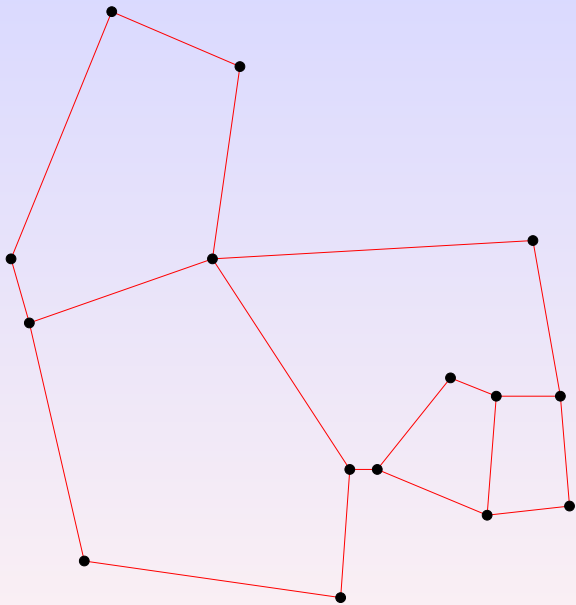
- find a good skeleton of the graph;
- decrease the size of the graph while preserving distances;
- optimize stretch-size tradeoffs.



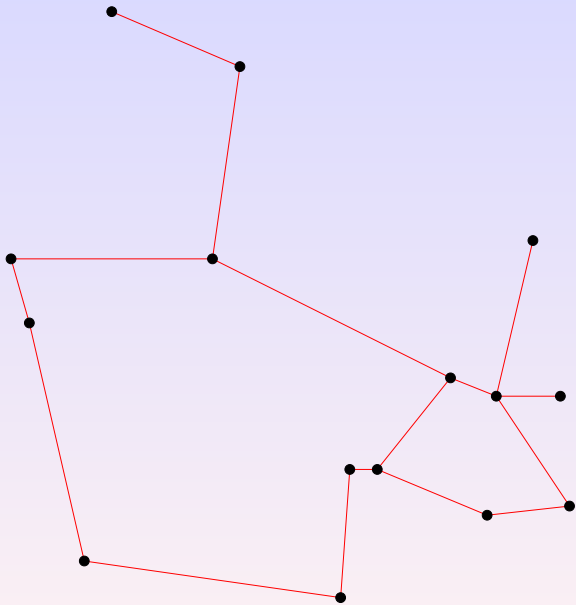
A complete Euclidian graph on 15 nodes



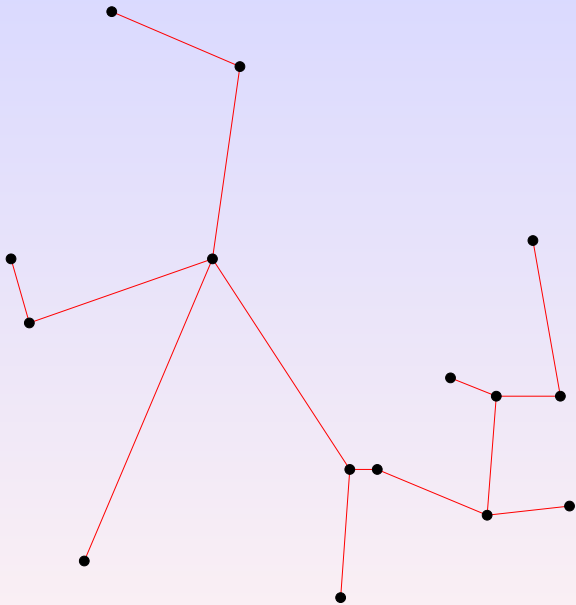
A (minimum cost) spanner with stretch 1.2



A (minimum cost) spanner with stretch 1.7



A (minimum cost) spanner with stretch 2.0



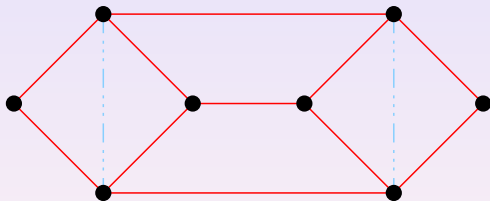
A (minimum cost) spanner with stretch 3.0

Definition

Definition

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

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

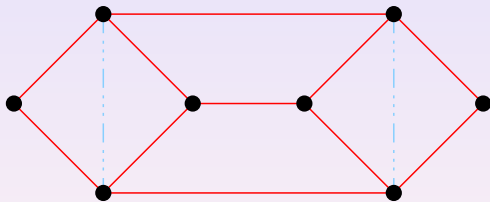


Definition

Definition

An (α, β) -spanner S of G is spanner of G satisfying $d_S(x, y) \leq \alpha \cdot d_G(x, y) + \beta$ 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”*

Used for:

- communication networks
- distributed systems
- network design

Spanners to do What?

Formally introduced by [Peleg-Ullman '87]: *“An optimal synchronizer for the Hypercube”*

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

Introduction, Examples & Definitions

On the Combinatorial Problem

Tree Cover Algorithms for $k = 2$

A Distributed Algorithm

Conclusion

Extremal Graph Theory

(Bollobàs, Bondy, Erdős, Simonovits, ...)

We consider (unweighted) n -node graphs.

Theorem (Folklore)

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

Extremal Graph Theory

(Bollobàs, Bondy, Erdős, Simonovits, ...)

We consider (unweighted) n -node graphs.

Theorem (Folklore)

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

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

Greedy Algorithm [Althöfer et al. '93]

Theorem

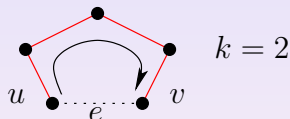
Every graph has a $(2k - 1, 0)$ -spanner with $O(n^{1+1/k})$ edges.

Greedy Algorithm [Althöfer et al. '93]

Theorem

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

- 1 $S := \emptyset$ (the empty graph)
- 2 While $\exists e = (u, v) \in E(G)$ with $d_S(u, v) > 2k - 1$,
 $S := S \cup \{e\}$

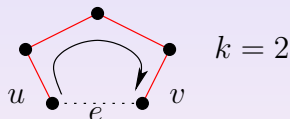


Greedy Algorithm [Althöfer et al. '93]

Theorem

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

- 1 $S := \emptyset$ (the empty graph)
- 2 While $\exists e = (u, v) \in E(G)$ with $d_S(u, v) > 2k - 1$,
 $S := S \cup \{e\}$



Properties

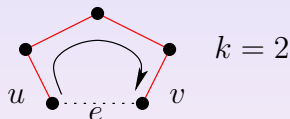
- $\text{stretch}(S) \leq 2k - 1$

Greedy Algorithm [Althöfer et al. '93]

Theorem

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

- 1 $S := \emptyset$ (the empty graph)
- 2 While $\exists e = (u, v) \in E(G)$ with $d_S(u, v) > 2k - 1$,
 $S := S \cup \{e\}$



Properties

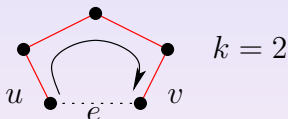
- $\text{stretch}(S) \leq 2k - 1$
- Whenever $e = (u, v)$ is added to S , one cannot create any cycle of length $\leq 2k$

Greedy Algorithm [Althöfer et al. '93]

Theorem

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

- 1 $S := \emptyset$ (the empty graph)
- 2 While $\exists e = (u, v) \in E(G)$ with $d_S(u, v) > 2k - 1$,
 $S := S \cup \{e\}$



Properties

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

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

Erdős-Simonovits Conjecture

Theorem [Folk] is optimal. More precisely,

Conjecture (ES82)

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

Erdős-Simonovits Conjecture

Theorem [Folk] is optimal. More precisely,

Conjecture (ES82)

For each integer $k \geq 1$, 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 $\geq 2k + 2$ with $\frac{1}{2}c_0 \cdot n^{1+1/k}$ edges.

Erdős-Simonovits Conjecture

Theorem [Folk] is optimal. More precisely,

Conjecture (ES82)

For each integer $k \geq 1$, 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 $\geq 2k + 2$ with $\frac{1}{2}c_0 \cdot n^{1+1/k}$ edges.

$B_1 =$ complete bipartite.

Erdős-Simonovits Conjecture

Theorem [Folk] is optimal. More precisely,

Conjecture (ES82)

For each integer $k \geq 1$, 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 $\geq 2k + 2$ 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 $\geq 2k + 2$ with $\Omega(n^{1+\frac{2}{3k}})$ edges.

Erdős-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 $|E(B_k)| = \Omega(n^{1+1/k})$.

Erdős-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 $\text{stretch}(u, v) \geq 2k + 1$.
But $\text{stretch}(u, v) \leq \alpha \cdot d_G(u, v) + \beta = \alpha + \beta$, implies
 $\alpha + \beta \geq 2k + 1$: \perp)

Erdős-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 $\text{stretch}(u, v) \geq 2k + 1$.
But $\text{stretch}(u, v) \leq \alpha \cdot d_G(u, v) + \beta = \alpha + \beta$, implies
 $\alpha + \beta \geq 2k + 1$: \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 $k \geq 1$, 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

Introduction, Examples & Definitions

On the Combinatorial Problem

Tree Cover Algorithms for $k = 2$

A Distributed Algorithm

Conclusion

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

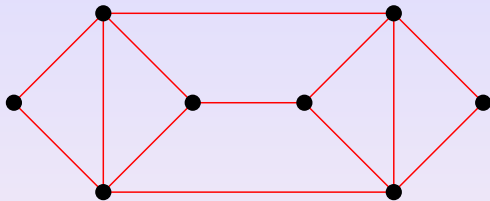
(stretch < 3 implies $\Omega(n^2)$ edges, and stretch < 5 implies $\Omega(n^{3/2})$ edges)

$B(u, r)$ = radius- r ball centered at u (in G)

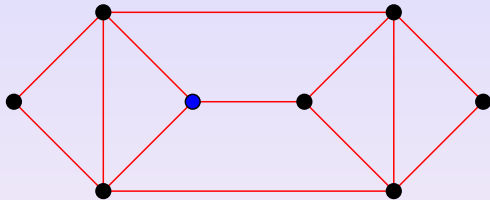
$\text{BFS}(u, X)$ = BFS tree rooted at u spanning X

- 1 $S := \emptyset$
- 2 While $\exists u \in V(G)$, $\deg(u) \geq \sqrt{n}$:
 - 1 $S := S \cup \text{BFS}(u, B(u, 2))$
 - 2 $G := G \setminus B(u, 1)$
- 3 $S := S \cup G$

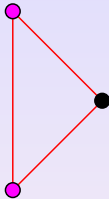
Example: $n = 8$, $\sqrt{n} \approx 2.82$



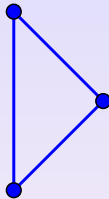
Example: $n = 8$, $\sqrt{n} \approx 2.82$



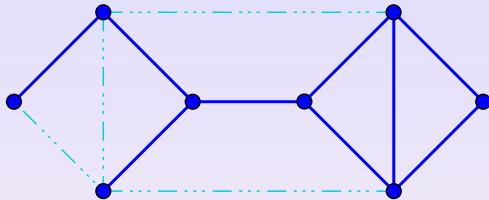
Example: $n = 8$, $\sqrt{n} \approx 2.82$



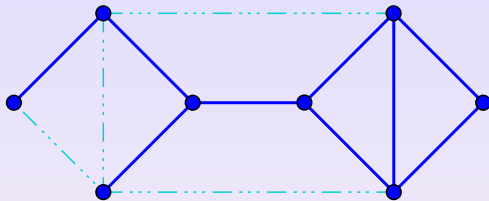
Example: $n = 8$, $\sqrt{n} \approx 2.82$



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

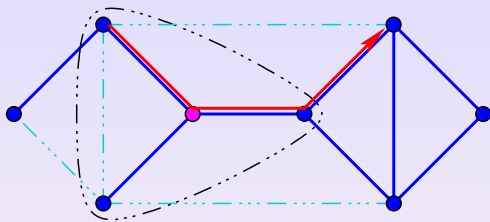


Example: $n = 8$, $\sqrt{n} \approx 2.82$



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

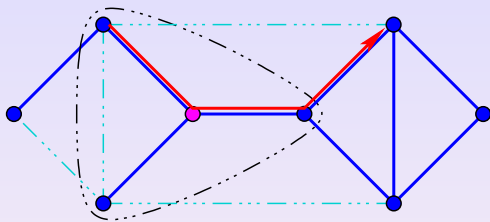
Example: $n = 8$, $\sqrt{n} \approx 2.82$



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

Stretch: 3

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



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

Stretch: 3

Note: It's a Tree Cover.

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

(Aingworth et al. '99)

- 1 $S := \emptyset$
- 2 While $\exists u \in V(G), \deg(u) \geq \sqrt{n}$:
 - 1 $S := S \cup \text{BFS}(u, G)$
 - 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 $S := \emptyset$
- 2 While $\exists u \in V(G), \deg(u) \geq \sqrt{n}$:
 - 1 $S := S \cup \text{BFS}(u, G)$
 - 2 $G := G \setminus B(u, 1)$
- 3 $S := S \cup G$

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

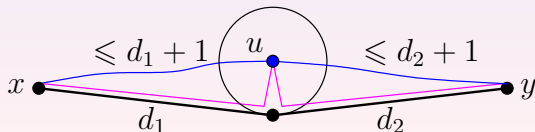
A $(1, 2)$ -spanners of size $O(n^{3/2})$

(Aingworth et al. '99)

- 1 $S := \emptyset$
- 2 While $\exists u \in V(G), \deg(u) \geq \sqrt{n}$:
 - 1 $S := S \cup \text{BFS}(u, G)$
 - 2 $G := G \setminus B(u, 1)$
- 3 $S := S \cup G$

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

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



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

Introduction, Examples & Definitions

On the Combinatorial Problem

Tree Cover Algorithms for $k = 2$

A Distributed Algorithm

Conclusion

The Model

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

- synchronic
- 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)

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 unknown, 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 unknown 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:

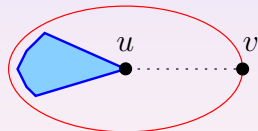
- 1 $R_u := \{u\} \cup \{ \text{any selection of } \lfloor \sqrt{|B(u, 3)|} \rfloor \text{ neighbors} \}$
- 2 send R_u and receive R_v to/from its neighbors
- 3 $W := B(u, 1) \setminus \{v \mid u \in R_v\} \setminus R_u$ and $C_u := \emptyset$
- 4 while $\exists w \in W$:
 - 1 $C_u := C_u \cup \{w\}$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

The Algorithm (for $k = 2$)

For every node u do:

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

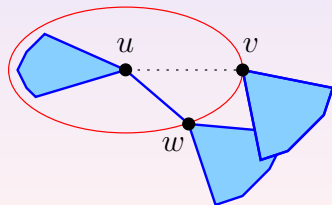


The Algorithm (for $k = 2$)

For every node u do:

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

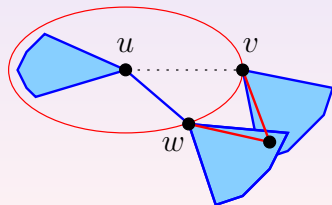


The Algorithm (for $k = 2$)

For every node u do:

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

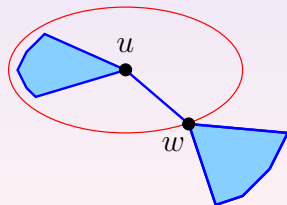


The Algorithm (for $k = 2$)

For every node u do:

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

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

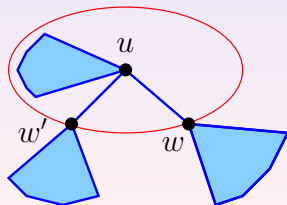


The Algorithm (for $k = 2$)

For every node u do:

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

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



Outline

Introduction, Examples & Definitions

On the Combinatorial Problem

Tree Cover Algorithms for $k = 2$

A Distributed Algorithm

Conclusion

Conclusion

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