A PTAS for the Sparsest Spanners Problem on Apex-Minor-Free Graphs

Feodor F. Dragan[1] Fedor V. Fomin[2] Petr A. Golovach[2]

[1]Department of Computer Science, Kent State University

[2]Department of Informatics, University of Bergen

The 33rd International Symposium on Mathematical Foundations of Computer Science, Toruń, 2008

Outline

1. **Introduction**
   - Multiplicative spanners
   - History and related work
   - Our results
2. **A PTAS for the Sparsest Spanners Problem**
   - Graphs of bounded local treewidth
   - Partial spanners
   - Idea of the algorithm
   - Apex-minor-free graphs

t-spanners

Definition (t-spanner)

Let t be a positive integer. A subgraph S of G, such that $V(S) = V(G)$, is called a *(multiplicative) t-spanner*, if $\mathrm{dist}_S(u, v) \leq t \cdot \mathrm{dist}_G(u, v)$ for every pair of vertices u and v. The parameter t is called the *stretch factor* of S.

t-spanners

Definition (t-spanner)

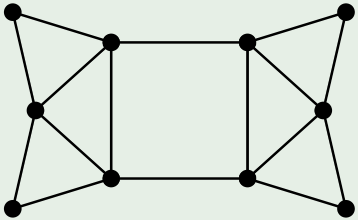
Let t be a positive integer. A subgraph S of G, such that $V(S) = V(G)$, is called a *(multiplicative) t-spanner*, if $\mathrm{dist}_S(u, v) \leq t \cdot \mathrm{dist}_G(u, v)$ for every pair of vertices u and v. The parameter t is called the *stretch factor* of S.

Observation (t-spanner)

Let G be a connected graph, and t be a positive integer. A spanning subgraph S of G is a t-spanner of G if and only if for every edge (x, y) of G, $\mathrm{dist}_S(x, y) \leq t$.

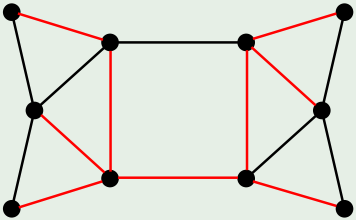
Examples of spanners

3 and 2-spanners



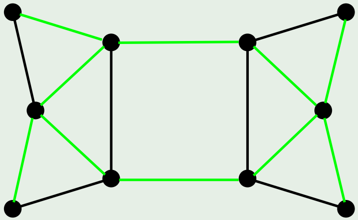
Examples of spanners

3 and 2-spanners



Examples of spanners

3 and 2-spanners



History and related work

- D. Peleg and J. D. Ullman, 1987

History and related work

- D. Peleg and J. D. Ullman, 1987
- D. Peleg and A. A. Schäffer, 1989: It is NP-complete to decide whether there is a t-spanner with at most m edges.

History and related work

- D. Peleg and J. D. Ullman, 1987
- D. Peleg and A. A. Schäffer, 1989: It is NP-complete to decide whether there is a t-spanner with at most m edges.
- G. Kortsarz, 2001: For every $t \geq 2$, there is a constant $c < 1$ such that it is NP-hard to approximate the sparsest t-spanner with the ratio $c \cdot \log n$.

History and related work

- D. Peleg and J. D. Ullman, 1987
- D. Peleg and A. A. Schäffer, 1989: It is NP-complete to decide whether there is a t-spanner with at most m edges.
- G. Kortsarz, 2001: For every $t \geq 2$, there is a constant $c < 1$ such that it is NP-hard to approximate the sparsest t-spanner with the ratio $c \cdot \log n$.
- U. Brandes and D. Handke, 1998: For every $t \geq 5$, it is NP-complete to decide whether there is a t-spanner with at most m edges for planar graphs.

History and related work

- D. Peleg and J. D. Ullman, 1987
- D. Peleg and A. A. Schäffer, 1989: It is NP-complete to decide whether there is a t-spanner with at most m edges.
- G. Kortsarz, 2001: For every $t \geq 2$, there is a constant $c < 1$ such that it is NP-hard to approximate the sparsest t-spanner with the ratio $c \cdot \log n$.
- U. Brandes and D. Handke, 1998: For every $t \geq 5$, it is NP-complete to decide whether there is a t-spanner with at most m edges for planar graphs.
- F. F. Dragan, F. V. Fomin and P. A. Golovach, 2008: For any fixed t and nonnegative integer r, it is possible to decide in a polynomial time whether an apex-minor-free graph G has a t-spanner with at most $n - 1 + r$ edges.

History and related work

- D. Peleg and J. D. Ullman, 1987
- D. Peleg and A. A. Schäffer, 1989: It is NP-complete to decide whether there is a t-spanner with at most m edges.
- G. Kortsarz, 2001: For every $t \geq 2$, there is a constant $c < 1$ such that it is NP-hard to approximate the sparsest t-spanner with the ratio $c \cdot \log n$.
- U. Brandes and D. Handke, 1998: For every $t \geq 5$, it is NP-complete to decide whether there is a t-spanner with at most m edges for planar graphs.
- F. F. Dragan, F. V. Fomin and P. A. Golovach, 2008: For any fixed t and nonnegative integer r, it is possible to decide in a polynomial time whether an apex-minor-free graph G has a t-spanner with at most $n - 1 + r$ edges.
- W. Duckworth, N. C. Wormald, and M. Zito, 2003: A PTAS for the sparsest 2-spanner problem on 4-connected planar triangulations.

Sparsest spanners for planar graphs

Problem (Sparsest t-spanner)

The SPARSEST t-SPANNER *problem asks to find, for a given graph G and an integer t, a t-spanner of G with the minimum number of edges.*

Sparsest spanners for planar graphs

Problem (Sparsest t-spanner)

The SPARSEST t-SPANNER *problem asks to find, for a given graph G and an integer t, a t-spanner of G with the minimum number of edges.*

Theorem (PTAS)

For every $t \geq 1$, the SPARSEST t-SPANNER *problem admits a PTAS with linear running time for the class of apex-minor-free graphs (and, hence, for the planar graphs and for the graphs with bounded genus).*

Graphs of bounded local treewidth

Definition (Graphs of bounded local treewidth)

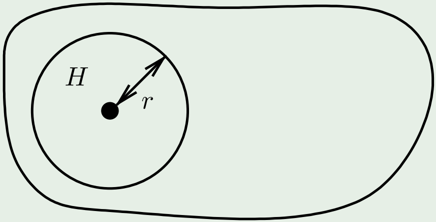
A graph class $\mathcal{G}$ has bounded local treewidth if there is function $f(r)$ (which depends only on r) such that for any graph G in $\mathcal{G}$, the treewidth of the subgraph of G induced by the set of vertices at distance at most r from any vertex is bounded above by $f(r)$.

Graphs of bounded local treewidth

Definition (Graphs of bounded local treewidth)

A graph class $\mathcal{G}$ has bounded local treewidth if there is function $f(r)$ (which depends only on r) such that for any graph G in $\mathcal{G}$, the treewidth of the subgraph of G induced by the set of vertices at distance at most r from any vertex is bounded above by $f(r)$.

Graphs of bounded local treewidth



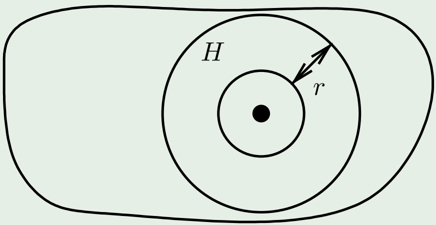
$tw(H) \leq f(r)$

Graphs of bounded local treewidth

Definition (Graphs of bounded local treewidth)

A graph class $\mathcal{G}$ has bounded local treewidth if there is function $f(r)$ (which depends only on r) such that for any graph G in $\mathcal{G}$, the treewidth of the subgraph of G induced by the set of vertices at distance at most r from any vertex is bounded above by $f(r)$.

Graphs of bounded local treewidth



$tw(H) \leq f(r)$

Partial spanners

Definition (Partial t-spanners)

Let $A \subseteq E(G)$. We call a subgraph S of G, such that for every edge $(x, y) \in A$ $\mathrm{dist}_S(x, y) \le t$, a *partial t-spanner for A*.

Partial spanners

Definition (Partial t-spanners)

Let $A \subseteq E(G)$. We call a subgraph S of G, such that for every edge $(x, y) \in A$ $\mathrm{dist}_S(x, y) \leq t$, a *partial t-spanner for A*.

Lemma

Let k and t be positive integers. Let also G be a graph of treewidth at most k, and let $A \subseteq E(G)$. The SPARSEST PARTIAL t-SPANNER *problem can be solved by a linear-time algorithm (the constant which is used in the bound of the running time depends only on k and t) if a corresponding tree decomposition of G is given.*

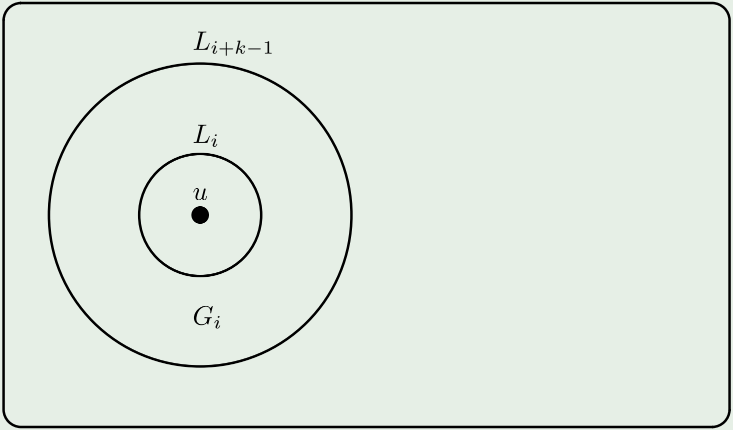
Idea of the algorithm

Definition (BFS decomposition)

Let u be a vertex of a graph G. For $i \geq 0$ we denote by L_i the i-th level of breadth first search, i.e. the set of vertices at distance i from u. We call the partition of the vertex set $V(G)$ $\mathcal{L}(G, u) = \{L_0, L_1, \dots, L_r\}$ *breadth first search (BFS) decomposition* of G.

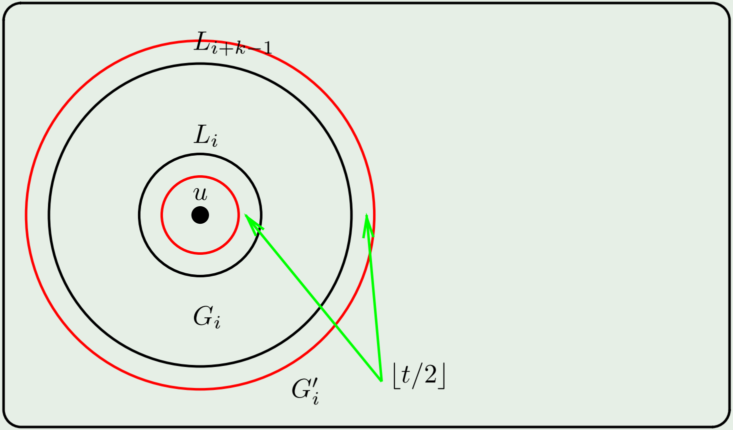
Idea of the algorithm

Graphs G_i and S_i



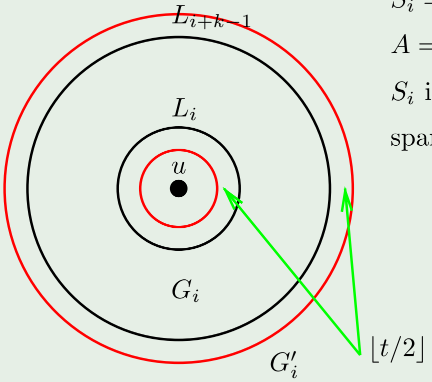
Idea of the algorithm

Graphs G_i and S_i



Idea of the algorithm

Graphs G_i and S_i



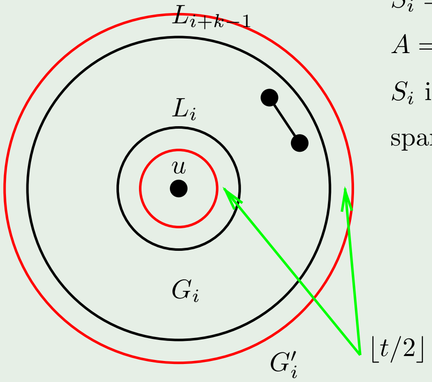
$S_i = S[V(G'_i)]$

$A = E(G_i)$

S_i is a partial spanner for A

Idea of the algorithm

Graphs G_i and S_i



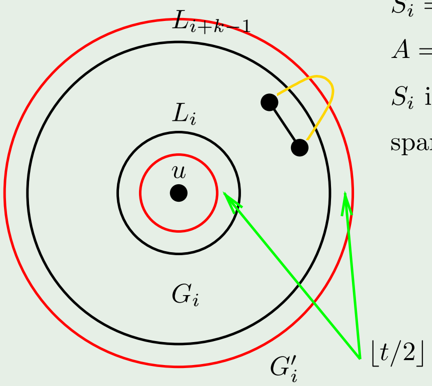
$S_i = S[V(G'_i)]$

$A = E(G_i)$

S_i is a partial spanner for A

Idea of the algorithm

Graphs G_i and S_i



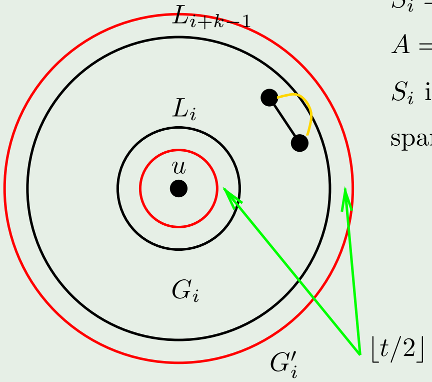
$S_i = S[V(G'_i)]$

$A = E(G_i)$

S_i is a partial spanner for A

Idea of the algorithm

Graphs G_i and S_i



$S_i = S[V(G'_i)]$

$A = E(G_i)$

S_i is a partial spanner for A

Idea of the algorithm

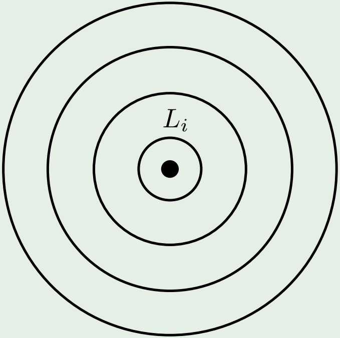
Lemma

Let S be a sparsest t-spanner of G, $S_i = S[V(G_i')]$. If S_i' is a sparsest partial t-spanner in G_i' for $A = E(G_i)$ then $|E(S_i')| \leq |E(S_i)|$.

Idea of the algorithm

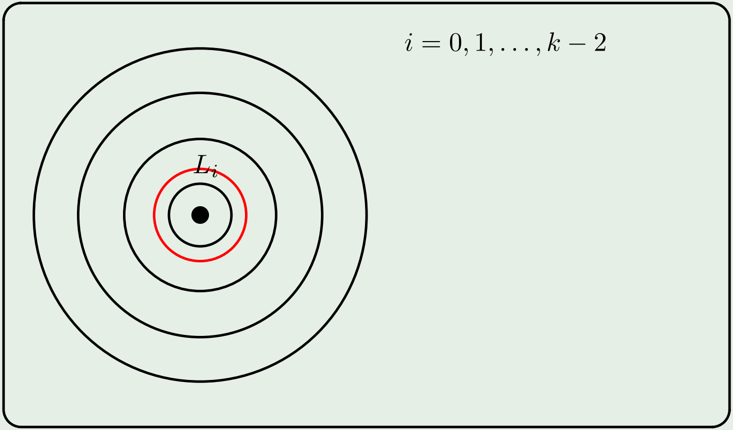
Construction of a spanner

$i = 0, 1, \ldots, k-2$



Idea of the algorithm

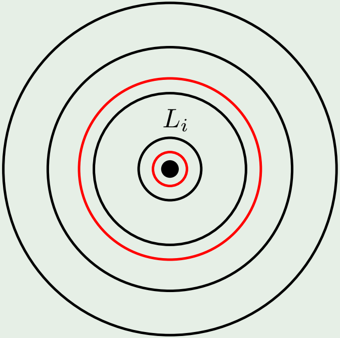
Construction of a spanner



Idea of the algorithm

Construction of a spanner

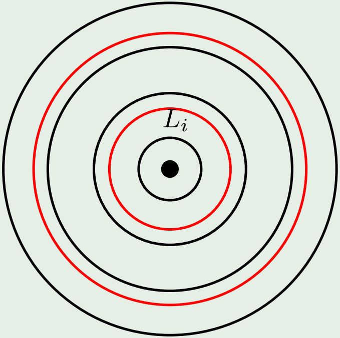
$i = 0, 1, \ldots, k-2$



Idea of the algorithm

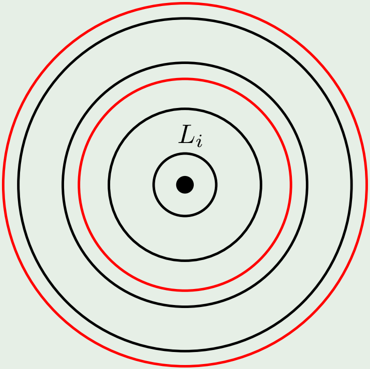
Construction of a spanner

$i = 0, 1, \ldots, k-2$



Idea of the algorithm

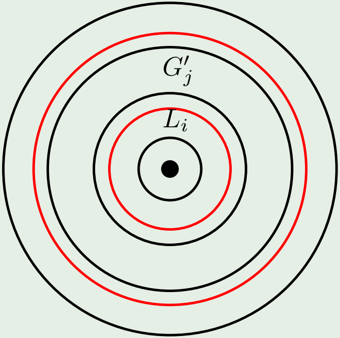
Construction of a spanner



$i = 0, 1, \ldots, k-2$

Idea of the algorithm

Construction of a spanner



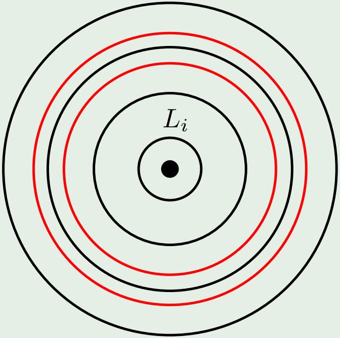
$i = 0, 1, \ldots, k-2$

S'_j - partial spanner
for $A = E(G_j)$

$S' = \cup S'_j$

Idea of the algorithm

Construction of a spanner



$i = 0, 1, \ldots, k-2$

$|E(S'_j)| \leq |E(S_j)|$

Idea of the algorithm

Lemma

The spanner S' has at most $(1 + \frac{t+1}{k-1})\mathrm{OPT}(G)$ edges, where $\mathrm{OPT}(G)$ is the number of edges in the solution of the SPARSEST *t-*SPANNER *problem on G.*

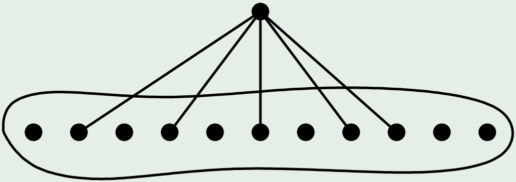
Definition (Apex graphs)

An *apex graph* is a graph obtained from a planar graph G by adding a vertex and making it adjacent to some vertices of G.

Definition (Apex graphs)

An *apex graph* is a graph obtained from a planar graph G by adding a vertex and making it adjacent to some vertices of G.

Apex graphs



Definition (Apex graphs)

An *apex graph* is a graph obtained from a planar graph G by adding a vertex and making it adjacent to some vertices of G.

Definition (Apex-minor-free graphs)

A graph class $\mathcal{G}$ is *apex-minor-free* if $\mathcal{G}$ excludes a fixed apex graph H as a minor.

Definition (Apex graphs)

An *apex graph* is a graph obtained from a planar graph G by adding a vertex and making it adjacent to some vertices of G.

Definition (Apex-minor-free graphs)

A graph class $\mathcal{G}$ is *apex-minor-free* if $\mathcal{G}$ excludes a fixed apex graph H as a minor.

Theorem (Eppstein)

All minor-closed graph classes that have bounded local treewidth are exactly apex-minor-free graph classes.

Thank you!