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

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

[1] Department of Computer Science, Kent State University, Kent, Ohio 44242, USA
dragan@cs.kent.edu
[2] Department of Informatics, University of Bergen, PB 7803, 5020 Bergen, Norway
{fedor.fomin,petr.golovach}@ii.uib.no

Abstract. A *t-spanner* of a graph G is a spanning subgraph S in which the distance between every pair of vertices is at most t times their distance in G. 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. On general n-vertex graphs, the problem is known to be NP-hard for all $t \geq 2$, and, even more, it is NP-hard to approximate it with ratio $O(\log n)$ for every $t \geq 2$. For $t \geq 5$, the problem remains NP-hard for planar graphs, and up to now the approximability status of the problem on planar graphs considered to be open. In this note, we resolve this open issue by showing that the SPARSEST t-SPANNER problem admits a *polynomial time approximation scheme (PTAS)* for every $t \geq 1$. Actually, our results hold for a much wider class of graphs, namely, on the class of *apex-minor-free graphs* which contains the classes of planar and bounded genus graphs.

1 Introduction

The concept of *sparse graph spanners* was introduced in [28] and [29] and has been studied since then in a number of papers, in the context of wired or wireless communication networks, distributed computing, robotics, computational geometry and biology [2,3,11,12,13,15,28,29]. A *t-spanner* of a graph G is a spanning subgraph S in which the distance between every pair of vertices is at most t times their distance in G. One is interested in finding a sparsest t-spanner for a graph G, i.e., a t-spanner with the minimum number of edges.

The original application of spanners was in the efficient simulation of synchronized protocols in unsynchronized networks [5,29]. Thereafter spanners were used in the design of low-stretch routing schemes using small routing tables (see [6,30] and the references therein), computing almost shortest paths in graphs [18], and in approximation algorithms for geometric spaces [27]. A recent application of spanners is in the design of approximate distance oracles and labeling schemes for arbitrary metrics; see [30,31] for further references. In all the applications cited above the quality of the solution is directly related to the quality of the

* Supported by Norwegian Research Council.

E. Ochmański and J. Tyszkiewicz (Eds.): MFCS 2008, LNCS 5162, pp. 290–298, 2008.

underlying spanners. For example, in [29], close relationships were established between the quality of spanners (in terms of *stretch factor* t and the number of spanner edges), and the time and communication complexities of any synchronizer for the network based on this spanner.

Unfortunately, as it was shown in [28], the problem of determining, for a given graph G and integers t and m, whether G has a t-spanner with at most m edges is NP-complete. This indicates that it is unlikely to find in polynomial time an exact solution for the sparsest t-spanner problem in general graphs even for small values of t and m. Later, [24] showed that 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 $c \cdot \log n$, where n is the number of vertices in the graph. On the other hand, the problem admits a $O(\log n)$-ratio approximation for $t = 2$ [25,24] and a $O(n^{2/(t+1)})$-ratio approximation for $t > 2$ [21]. For some other inapproximability and approximability results for the sparsest t-spanner problem on general graphs we refer the reader to [19,20,21] and papers cited therein.

In this note, we consider the sparsest t-spanner problem on so-called *apex-minor-free graphs* which is a large class of graphs including all planar graphs and all graphs with bounded genus. Spanners for these graph classes were considered in [16]. Particularly, it was shown that for any fixed positive integer t and nonnegative integer r, it is possible to decide in a polynomial time whether a graph G has a t-spanner with at most $n-1+r$ edges. From another side, it is known that, on planar graphs, the problem of determining, for a given graph G and integers m and t, if G has a t-spanner with at most m edges is NP-complete for every fixed $t \geq 5$ (the case $2 \leq t \leq 4$ is open) [10]. This indicates that it is unlikely to find in polynomial time an exact solution for the sparsest t-spanner problem in planar graphs, too, and, consequently, a possible remaining course of action for investigating the problem is devising approximation algorithms for it.

Here, we show that the sparsest t-spanner problem admits a polynomial time approximation scheme (PTAS) on the class of apex-minor-free graphs for every $t \geq 1$ (and, hence, for the planar graphs and for the graphs with bounded genus). For NP-hard optimization problems, a PTAS is one of the best types of algorithm one can hope for. In proving our result, we employ the well known technique for solving NP-hard problems on planar graphs proposed by Baker [7] and generalized by Eppstein [22,23] (see also [14]) to graphs with bounded local treewidth (alias, apex-minor-free graphs). Previously, a PTAS was known only for the sparsest 2-spanner problem on 4-connected planar triangulations [17].

Our result also answers the following questions explicitly mentioned in in [10] and [17]:

– What is the approximability status of the sparsest t-spanner problem for planar graphs?
– Does a PTAS exist for the sparsest t-spanner problem for 4-connected planar triangulations and $t > 2$, or even for all planar graphs?

2 Preliminaries

In this section we present necessary definitions, notations and some auxiliary results.

Let $G = (V, E)$ be an undirected graph with the vertex set V and edge set E. We often will use notations $V(G) = V$ and $E(G) = E$. For $U \subseteq V$ by $G[U]$ is denoted the subgraph of G induced by U. The *distance* $\mathrm{dist}_G(u, v)$ between vertices u and v of a connected graph G is the length (the number of edges) of a shortest u, v-path in G.

Let t be a positive integer. A subgraph S of G, such that $V(S) = V(G)$, is called a *(multiplicative) t-spanner* of G, 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. It is easy to see that the t-spanners can equivalently be defined as follows.

Proposition 1. *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 $\mathrm{dist}_S(x, y) \leq t$.*

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*. Clearly, if $A = E(G)$ then a partial t-spanner for this set is a t-spanner for G.

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. Correspondingly, the SPARSEST PARTIAL t-SPANNER problem asks to find a partial t-spanner with the minimum number of edges for a given graph G, an integer t and a set $A \subset E(G)$.

A *tree decomposition* of a graph G is a pair (X, U) where U is a tree whose vertices we call *nodes* and $X = (\{X_i \mid i \in V(U)\})$ is a collection of subsets of $V(G)$ such that

1. $\bigcup_{i \in V(U)} X_i = V(G)$,
2. for each edge $(v, w) \in E(G)$, there is an $i \in V(U)$ such that $v, w \in X_i$, and
3. for each $v \in V(G)$ the set of nodes $\{i \mid v \in X_i\}$ forms a subtree of U.

The *width* of a tree decomposition $(\{X_i \mid i \in V(U)\}, U)$ equals $\max_{i \in V(U)} \{|X_i| - 1\}$. The *treewidth* of a graph G is the minimum width over all tree decompositions of G. We use notation $\mathbf{tw}(G)$ to denote the treewidth of a graph G.

It is said that a graph class $\mathcal{G}$ has bounded local treewidth if there is a 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 $f(r)$. A graph class $\mathcal{G}$ has *linear local treewidth* if $f(r) = O(r)$. For example, it is known [9,1] that, for every planar graph G, $f(r) \leq 3r - 1$, and a corresponding tree decomposition of width at most $3r - 1$ of the subgraph induced by the set of vertices at distance at most r from any vertex can be found in time $O(rn)$.

Given an edge $e = (x, y)$ of a graph G, the graph G/e is obtained from G by contracting the edge e; that is, to get G/e we identify the vertices x and y

and remove all loops and replace all multiple edges by simple edges. A graph H obtained by a sequence of edge-contractions is said to be a *contraction* of G. H is a *minor* of G if H is a subgraph of a contraction of G. A graph class $\mathcal{G}$ is *minor-closed* if for every graph $G \in \mathcal{G}$ all minors of G are in $\mathcal{G}$, too.

We say that a graph G is *H-minor-free* when it does not contain H as a minor. We also say that a graph class $\mathcal{G}$ is *H-minor-free* (or, excludes H as a minor) when all its members are H-minor-free. Clearly, all minor-free graph classes are minor-closed.

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. A graph class is *apex-minor-free* if it does not contain any graph with some fixed apex graph as a minor. For example, planar graphs (and bounded-genus graphs) are apex-minor-free graphs.

Eppstein [22,23] characterized all minor-closed graph classes that have bounded local treewidth. It was proved that they are exactly apex-minor-free graphs. These results were improved by Demaine and Hajiaghayi [14]. They proved that all apex-minor-free graphs have linear local treewidth.

3 Main Result

Many optimization problems can be solved efficiently for graphs of bounded treewidth by formulating the problem in a logical language, called *Monadic Second Order Logic* (abbr. MSOL). It is known that problems which can be expressed in this way can be solved in linear time for graphs with bounded treewidth [4]. We need such a result for the SPARSEST PARTIAL t-SPANNER problem.

Lemma 1. *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.*

Proof. The SPARSEST PARTIAL t-SPANNER problem can formulated in MSOL as follows. We ask for a a subgraph S of G (i.e. a subset of edges) with the following property: for every edge $(x, y) \in A$ $\text{dist}_S(x, y) \leq t$. This property is expressible in MSOL because $\text{dist}_S(x, y) \leq t$ means that there are edges $(v_0, v_1), (v_1, v_2), \ldots, (v_{l-1}, v_l) \in E(S)$ for some $l \leq t$ such that $x = v_0$ and $y = v_l$. Then the claim follows from the well known results of Arnborg et al.[4]. □

It should be noted also that the dynamic-programming algorithm for the case $A = E(G)$ was given by Makowsky and Rotics [26]. The algorithm of Makowsky and Rotics can be easily adapted to solve the problem for arbitrarily choice of A.

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 $V(G)$ $\mathcal{L}(G, u) = \{L_0, L_1, \ldots, L_r\}$ *breadth first search (BFS) decomposition* of G. We assume for convenience that for BFS decomposition $\mathcal{L}(G, u)$ $L_i = \emptyset$ for $i < 0$ or $i > r$, and we use further negative indices and indices that are more than r. It can be easily seen that the BFS decomposition can be constructed by the breadth first search in a linear time.

Let G be a graph with BFS decomposition $\mathcal{L}(G,u) = (L_0, L_1, \ldots, L_r)$, and t be a positive integer. Suppose that $i \le j$ are integers. For $i \le j$ we define

$$G_{ij} = G[\bigcup_{k=i}^{j} L_k].$$

Graph G_{ij} is shown on Fig. 1.

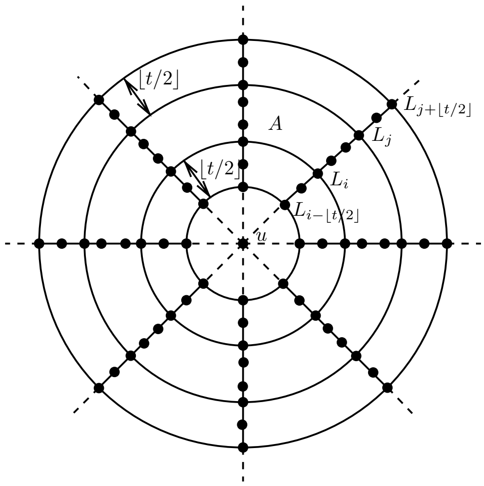

Fig. 1. Graphs G_{ij} and G'_{ij}

The following result is due to Demaine and Hajiaghayi [14] (see also the work of Eppstein [23]),

Lemma 2 ([14]). *Let G be an apex-minor-free graph. Then* $\mathbf{tw}(G_{ij}) = O(j-i)$.

Denote by $G'_{ij} = G_{i-\lfloor t/2 \rfloor, j+\lfloor t/2 \rfloor}$ (see Fig. 1), and let $A = E(G_{ij})$. Let S be a t-spanner of G and S' be the subgraph of S induced by $V(G'_{ij})$. We need the following claim.

Lemma 3. *S' is a partial t-spanner for A in G'_{ij}.*

Proof. Let $(x,y) \in A$. Note that $x,y \in V(G_{ij})$. Since S is a t-spanner for G, we have that there is a x,y-path P in S of length at most t. Suppose that some vertex v of this path does not belong to G'. Then $v \in L_l$ for some $l < i - \lfloor t/2 \rfloor$ or $l > j + \lfloor t/2 \rfloor$. By the definition of the BFS decomposition $\mathrm{dist}_G(x,v) > \lfloor t/2 \rfloor$ and $\mathrm{dist}_G(y,v) > \lfloor t/2 \rfloor$. But then P has length at least $\mathrm{dist}_G(x,v) + \mathrm{dist}_G(v,y) \ge 2\lfloor t/2 \rfloor + 2 > t$. So, all vertices of P are vertices of G'_{ij}, and this path is a path in S'. □

Now we are ready to describe our algorithm. Let t, k be positive integers, $t < k$. For a given apex-minor-free graph G the BFS decomposition $\mathcal{L}(G, u) = (L_0, L_1, \ldots, L_r)$ is constructed for some vertex u.

If $r \leq k$ then a t-spanner S of G is constructed directly. We use the fact that $\mathbf{tw}(G) = O(k)$ and, for example, use Bodlaender's Algorithm [8] to construct in linear time a suitable tree decomposition of G. Then, by Lemma 1, a sparsest t-spanner of G can be found in linear time.

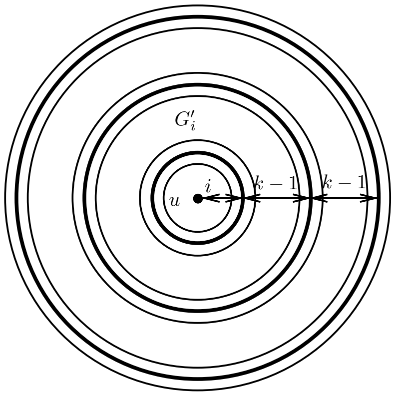

Fig. 2. Graphs G'_j

Suppose now that $r > k$. We consequently construct t-spanners S_i of G for $i = 1, 2, \ldots, k-1$ as follows. Let

$$J_i = \{j \in \{2-k, 3-k, \ldots, r-1\} \colon j \equiv i \pmod{k-1}\}.$$

For every $j \in J_i$ we consider graph $G'_j = G_{j-\lfloor t/2 \rfloor, j+k+\lfloor t/2 \rfloor -1}$ and set of edges $A_j = E(G_{j,j+k-1})$. In other words, we "cover" graph G by graphs $G'_{i-(k-1)}, G'_i, G'_{i+(k-1)}, \ldots$, and two consecutive graphs "overlap" by $2\lfloor t/2 \rfloor + 1$ levels in the BFS decomposition (see Fig. 2). The union of all sets A_j is the set $E(G)$. By Lemma 2, $\mathbf{tw}(G'_j) = O(k+t)$. For every graph G'_j we construct a sparsest partial t-spanner S_{ij} for A_j in G'_j by making use of Lemma 1. We define

$$S_i = \bigcup_{j \in J_i} S_{ij}.$$

Finally, we choose among graphs $S_1, S_2, \ldots, S_{k-1}$ the graph with the minimum number of edges and denote it by S.

The following theorem describes properties of the graph S.

Theorem 1. *Let S be the subgraph of an apex-minor-free graph G, obtained by the algorithm described above. Then the following holds*

1. *S is a t-spanner of G.*

2. *For every* t *and* $k > t$, S *can be constructed by a linear-time algorithm.*
3. 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.

Proof. 1. Every S_i is a t-spanner of G. Indeed, for every $(x, y) \in E(G)$, there is $j \in J_i$ such that $(x, y) \in A_j$, and $\mathrm{dist}_{S_i}(x, y) \leq \mathrm{dist}_{S_{ij}}(x, y) \leq t$.

2. The second claim yields by Lemmata 1 and 2.

3. If $k \geq r$ then the claim is obvious. Let $k < r$ and let T be a t-spanner of G with the minimum number of edges, $m = |E(T)| = \mathrm{OPT}(G)$. Assume that $i \in \{1, 2, \ldots, k-1\}$ and $j \in J_i$. Let $T_j = T[V(G'_j)]$. By Lemma 3, T_j is a partial t-spanner for the set A_j in T_j. Then

$$|E(T_j)| \geq |E(S_{ij})|,$$

and

$$\begin{aligned} |E(S_i)| &\leq \sum_{j \in J_i} |E(T_j)| \\ &= m + \sum_{j \in J_i} |E(T) \cap E(G_{j-\lfloor t/2 \rfloor, j+\lfloor t/2 \rfloor})|. \end{aligned}$$

We have only to note that

$$\begin{aligned} |E(S)| &= \min_{1 \leq i \leq k-1} |E(S_i)| \\ &\leq m + \min_{1 \leq i \leq k-1} \sum_{j \in J_i} |E(T) \cap E(G_{j-\lfloor t/2 \rfloor, j+\lfloor t/2 \rfloor})| \\ &\leq m + \min_{1 \leq i \leq k-1} \sum_{j \in J_i} |E(G_{j-\lfloor t/2 \rfloor, j+\lfloor t/2 \rfloor})| \\ &\leq (1 + \frac{t+1}{k-1})m. \end{aligned}$$ □

Finally, we have the following corollary.

Corollary 1. *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).*

Note that, since the proof of Lemma 1 was not constructive, we can not claim that we have a very efficient algorithm. It would be interesting to find a more efficient solution to the problem at least for the class of planar graphs by utilizing the dynamic programming technique, the planarity of the graph and the specifics of the problem. For the planar graphs, the initial problem on G will be reduced to a subproblem of constructing a sparsest partial t-spanner for a subgraph of G with bounded outerplanarity.

References

1. Alber, J., Bodlaender, H.L., Fernau, H., Kloks, T., Niedermeier, R.: Fixed parameter algorithms for dominating set and related problems on planar graphs. Algorithmica 33, 461–493 (2002)
2. Althöfer, I., Das, G., Dobkin, D., Joseph, D., Soares, J.: On sparse spanners of weighted graphs. Discrete Comput. Geom. 9, 81–100 (1993)
3. Alzoubi, K., Li, X.-Y., Wang, Y., Wan, P.J., Frieder, O.: Geometric spanners for wireless ad hoc networks. IEEE Transactions on Parallel and Distributed Systems 14, 408–421 (2003)
4. Arnborg, S., Lagergren, J., Seese, D.: Easy problems for tree-decomposable graphs. J. Algorithms 12, 308–340 (1991)
5. Awerbuch, B.: Complexity of network synchronization. J. ACM 32, 804–823 (1985)
6. Awerbuch, B., Peleg, D.: Routing with polynomial communication-space trade-off. SIAM Journal on Discrete Mathematics 5, 151–162 (1992)
7. Baker, B.S.: Approximation algorithms for NP-complete problems on planar graphs. J. Assoc. Comput. Mach. 41, 153–180 (1994)
8. Bodlaender, H.L.: A linear-time algorithm for finding tree-decompositions of small treewidth. SIAM J. Comput. 25, 1305–1317 (1996)
9. Bodlaender, H.L.: A partial k-arboretum of graphs with bounded treewidth. Theoretical Computer Science 209, 1–45 (1998)
10. Brandes, U., Handke, D.: NP-completeness results for minimum planar spanners. Discrete Mathematics and Theoretical Computer Science 3, 1–10 (1998)
11. Cai, L.: Tree spanners: Spanning trees that approximate the distances, Ph.D. thesis, University of Toronto (1992)
12. Cai, L., Corneil, D.G.: Tree spanners. SIAM J. Disc. Math. 8, 359–387 (1995)
13. Chew, L.P.: There are planar graphs almost as good as the complete graph. J. of Computer and System Sciences 39, 205–219 (1989)
14. Demaine, E.D., Hajiaghayi, M.T.: Fast Algorithms for Hard Graph Problems: Bidimensionality, Minors, and Local Treewidth. In: Pach, J. (ed.) GD 2004. LNCS, vol. 3383, pp. 517–533. Springer, Heidelberg (2005)
15. Dobkin, D.P., Friedman, S.J., Supowit, K.J.: Delaunay graphs are almost as good as complete graphs. Discrete and Computational Geometry 5, 399–407 (1990)
16. Dragan, F.F., Fomin, F.V., Golovach, P.A.: Spanners in sparse graphs. In: Aceto, L., et al. (eds.) ICALP 2008, Part I. LNCS, vol. 5125, pp. 597–608. Springer, Heidelberg (2008)
17. Duckworth, W., Wormald, N.C., Zito, M.: A PTAS for the sparsest 2-spanner of 4-connected planar triangulations. J. Discrete Algorithms 1, 67–76 (2003)
18. Elkin, M.: Computing almost shortest paths. ACM Transactions on Algorithms 1, 283–323 (2005)
19. Elkin, M., Peleg, D.: Strong Inapproximability of the Basic t-Spanner Problem. In: Montanari, U., Rolim, J.D.P., Welzl, E. (eds.) ICALP 2000. LNCS, vol. 1853, pp. 636–647. Springer, Heidelberg (2000)
20. Elkin, M., Peleg, D.: The Hardness of Approximating Spanner Problems. In: Reichel, H., Tison, S. (eds.) STACS 2000. LNCS, vol. 1770, pp. 370–381. Springer, Heidelberg (2000)
21. Elkin, M., Peleg, D.: Approximating k-spanner problems for $k > 2$. Theoret. Comput. Sci. 337, 249–277 (2005)
22. Eppstein, D.: Subgraph isomorphism in planar graphs and related problems. In: SODA, pp. 632–640 (1995)

23. Eppstein, D.: Diameter and treewidth in minor-closed graph families. Algorithmica 27, 275–291 (2000)
24. Kortsarz, G.: On the Hardness of Approximating Spanners. Algorithmica 30, 432–450 (2001)
25. Kortsarz, G., Peleg, D.: Generating Sparse 2-Spanners. J. Algorithms 17, 222–236 (1994)
26. Makowsky, J.A., Rotics, U.: Optimal spanners in partial k-trees (manuscript)
27. Narasimhan, G., Smid, M.: Geometric Spanner Networks. Cambridge (2007)
28. Peleg, D., Schäffer, A.A.: Graph spanners. J. Graph Theory 13, 99–116 (1989)
29. Peleg, D., Ullman, J.D.: An optimal synchronizer for the hypercube. SIAM J. Comput. 18, 740–747 (1989)
30. Thorup, M., Zwick, U.: Compact routing schemes. In: Proceedings of the 13th Ann. ACM Symp. on Par. Alg. and Arch (SPAA 2001), pp. 1–10. ACM Press, New York (2001)
31. Thorup, M., Zwick, U.: Approximate distance oracles. J. ACM 52, 1–24 (2005)