Information Processing Letters 111 (2011) 142–144

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Spanners of bounded degree graphs

Fedor V. Fomin [a], Petr A. Golovach [b,1], Erik Jan van Leeuwen [a,*]

[a] Department of Informatics, University of Bergen, PB 7803, N-5020 Bergen, Norway
[b] School of Engineering and Computing Sciences, Durham University, South Road, Durham DH1 3LE, UK

ARTICLE INFO

Article history:
Received 2 August 2010
Received in revised form 21 October 2010
Accepted 22 October 2010
Available online 29 October 2010
Communicated by Ł. Kowalik

Keywords:
Graph algorithms
Graph spanners
Fixed-parameter tractability
NP-hardness

ABSTRACT

A *k-spanner* of a graph G is a spanning subgraph of G in which the distance between any pair of vertices is at most k times the distance in G. We prove that for fixed k, w, the problem of deciding if a given graph has a k-spanner of treewidth w is fixed-parameter tractable on graphs of bounded degree. In particular, this implies that finding a k-spanner that is a tree (a *tree k-spanner*) is fixed-parameter tractable on graphs of bounded degree. In contrast, we observe that if the graph has only one vertex of unbounded degree, then TREE k-SPANNER is NP-complete for $k \geqslant 4$.

1. Introduction

Spanning subnetworks or *spanners* are very useful tools in streamlining communications in networks by way of a backbone. In designing a network spanner, one often has a trade-off between the complexity of the spanner and how well the spanner approximates the network. If we insist that the spanner is a tree, we therefore focus on ensuring that the length of a path between two endpoints increases minimally when using the spanner to route communications.

Peleg et al. [19,20] were the first to suggest this way of looking at spanners. Given a graph G and a spanning subgraph H of G, the *stretch factor* of an edge of G is the distance between its two endpoints in H. A *k-spanner* is a spanning subgraph of G in which none of the edges of G have stretch factor more than k. If the spanner must be a tree, then this is a *tree k-spanner*. In [20], close relationships were established between the quality of spanners (in terms of stretch factor and the number of spanner edges), and the time and communication complexities of any synchronizer for the network based on this spanner. Another example is the usage of tree k-spanners in the analysis of arrow distributed queuing protocols [12].

Substantial work has been done on the TREE k-SPANNER problem, also known as the minimum stretch spanning tree problem. Cai and Corneil [3] have shown that, for a given graph G, the problem to decide whether G has a tree k-spanner is NP-complete for any fixed $k \geqslant 4$ and is linear time solvable for $k = 1, 2$ (the status of the case $k = 3$ is open for general graphs). An $O(\log n)$-approximation algorithm for the minimum value of k for the TREE k-SPANNER problem is due to Emek and Peleg [8]. See the survey of Peleg [18] and some recent papers [1,7,9] for more details on this problem and its variants.

The TREE k-SPANNER problem on sparse graphs was studied intensively. Fekete and Kremer proved that the TREE k-SPANNER problem on planar graphs is NP-complete (when k is part of the input) (see [10] for the journal version). They also showed that it can be decided in polynomial time whether a given planar graph has a tree 3-spanner and left open the question for $k > 3$.

Dragan et al. [6] introduced spanners of bounded treewidth, motivated by the fact that many algorithmic problems are tractable on graphs of bounded treewidth,

* Corresponding author.
E-mail addresses: fedor.fomin@ii.uib.no (F.V. Fomin), petr.golovach@durham.ac.uk (P.A. Golovach), E.J.van.Leeuwen@ii.uib.no (E.J. van Leeuwen).

1 Supported by the EPSRC under project EP/G043434/1.

0020-0190/$ – see front matter
doi:10.1016/j.ipl.2010.10.021

and a spanner of small treewidth can be used to obtain an approximate solution to a problem on G. They proved that the problem of finding a k-spanner of treewidth at most w in a given planar graph G is fixed-parameter tractable parameterized by k and w, and for every fixed k and w, this problem can be solved in linear time. Moreover, they proved that this result extends from planar graphs to the more general class of apex-minor-free graphs. Also they showed that the tractability border of the k-Spanner problem cannot be extended beyond the class of apex-minor-free graphs, as for every $k \geqslant 4$, the Tree k-Spanner problem is NP-complete on K_6-minor-free graphs. It should be noted that the techniques used by Dragan et al. do not extend directly to other classes of sparse graphs, like graphs of bounded degree, and thus their results do not carry over.

In this note, we consider the problems of deciding whether a given graph has a tree k-spanner or a k-spanner of treewidth at most w. We show that on graphs of bounded degree, these problems can be solved in linear time for any fixed k, w. We complement this result by the observation that for any fixed $k \geqslant 4$, it is NP-complete to decide whether a graph G with at most one vertex of degree greater than 8 has a tree k-spanner.

2. Preliminaries

Throughout the paper, we use only undirected simple graphs. The degree of a vertex v of graph G is denoted by $\deg(v)$. The maximum degree of any vertex of G is denoted by $\Delta(G)$, and is called the *degree of G*.

We now define the notion of fixed-parameter tractability. We say that a problem is *fixed-parameter tractable* or in class *FPT*, if there is a computable function f such that the problem can be decided in $f(k) \cdot n^{O(1)}$ time, for any problem instance of size n and any parameter k. We refer to recent monographs of Flum and Grohe [11] and Niedermeier [15] for overviews of parameterized complexity.

Our results rely on tree decompositions of graphs. A *tree decomposition* (T, X) of a graph G is a tree T and a collection of bags $X_t \subseteq V(G)$, one for each vertex $t \in V(T)$, satisfying three conditions: (i) $\bigcup_{t \in V(T)} X_t = V(G)$, (ii) $\forall (u, v) \in E(G)$, there is a vertex $t \in V(T)$ such that $u, v \in X_t$, and (iii) $X_t \cap X_{t''} \subseteq X_{t'}$ for any $t, t'' \in V(T)$ and for any $t' \in V(T)$ on the t–t''-path in T. The *width* of a tree decomposition is $\max_{t \in V(T)}\{|X_t|\} - 1$. The *treewidth* $\mathrm{tw}(G)$ of a graph G is the minimum width of any tree decomposition of G.

3. Result

We first prove that deciding the existence of a tree k-spanner is fixed-parameter tractable on graphs of bounded degree.

We need a few auxiliary notions [16]. Let G be a graph and T a spanning tree of G. The *detour* for an edge $(u, v) \in E(G)$ is the u–v-path in T. The *congestion* of an edge $e \in E(T)$, denoted by $\mathrm{cng}_{G,T}(e)$, is the number of detours that contain e. The *congestion* of G w.r.t. T, denoted by $\mathrm{cng}_G(T)$, is the maximum congestion over all edges in T. The *spanning tree congestion* $\mathrm{stc}(G)$ is the minimum congestion over all spanning trees T of G.

Lemma 3.1. *Let T be a tree k-spanner of a graph G of degree $\Delta = \Delta(G)$. Then $\mathrm{stc}(G) \leqslant \mathrm{cng}_G(T) \leqslant 2 \cdot (\Delta - 1)^{k-1}$.*

Proof. Consider an arbitrary edge $e \in E(T)$. Since T is a k-spanner, all detours passing through e must be from edges within distance $k - 1$ of e. As G has degree Δ, the number of such edges is at most $2 \cdot (\Delta - 1)^{k-1}$. □

Corollary 3.2. *If G has a tree k-spanner, then $\mathrm{tw}(G) \leqslant (\Delta(G))^k$.*

Proof. This is immediate from Lemma 3.1 and the fact that $\mathrm{tw}(G) \leqslant \max\{\mathrm{stc}(G), \Delta(G) \cdot (\mathrm{stc}(G) - 1)/2\}$ (as proved by Otachi, Bodlaender, and van Leeuwen [17]). □

Theorem 3.3. *For fixed k, the Tree k-Spanner problem can be decided in linear time on graphs of bounded degree.*

Proof. Let G be a graph of degree $\Delta = \Delta(G)$. We use Bodlaender's algorithm [2] to decide in linear time whether $\mathrm{tw}(G) \leqslant \Delta^k$. If not, then G does not have a tree k-spanner by Corollary 3.2. Otherwise, we use that tree k-spanner is expressible in monadic second-order logic (MSOL) [6]. By applying Courcelle's theorem [4,5], which states that any MSOL problem can be decided in linear time on graphs of bounded treewidth, we can then decide the existence of a tree k-spanner in linear time. □

We now extend these results to spanners of bounded treewidth.

Lemma 3.4. *If G has a k-spanner H of treewidth w, then it holds that $\mathrm{tw}(G) \leqslant (w+1) \cdot (1 + (\Delta(G) - 1)^{k-1}) \cdot \max\{\Delta(G) + 4, w + 4\}$.*

Proof. Let $\Delta = \Delta(G)$ and let (T, X) be a tree decomposition of H of width w. It is known that one can assume that $\Delta(T) \leqslant 3$ [13]. Associate the graph $G_t = G[X_t]$ with each vertex $t \in V(T)$. Construct a new graph G' from the union of T and these G_t as follows. If $v \in X_t \cap X_{t'}$ for $(t, t') \in E(T)$, add an edge between the copy of v in G_t and the copy of v in $G_{t'}$. For any $(u, v) \in E(G - H)$, add an edge in G' between some copy of u and some copy of v. Finally, add an edge between each $t \in V(T)$ and each vertex of G_t. Denote the latter set of edges by F.

Clearly, the graph induced by F and T is a spanning tree T' of G'. The congestion of any edge in F is bounded by $\Delta + 4$. So consider any edge $e' = (t, t') \in E(T)$ and let $X = X_t \cap X_{t'}$. Obviously, e' is only in detours for edges $e \in E(G - H)$ and edges between the copies of vertices of X in G_t and $G_{t'}$. The latter amounts to at most $|X| \leqslant w + 1$ edges. To count the former, note that the detour in H of an edge $e \in E(G - H)$ is some path $P = p_1 \ldots p_\ell$ of length at most k. This path corresponds to a walk in T, visiting the bag containing p_i, p_{i+1} for each $i = 1, \ldots, \ell - 1$. Such a walk goes through e' only if a vertex of P is in X. As P has at most k edges, e must be within distance k of a vertex of X. Using the same arguments as in Lemma 3.1, it follows that the number of these detours going through e' is at most $(w+1) \cdot 2(\Delta-1)^{k-1}$. Hence $\mathrm{stc}_{G'}(T') \leqslant 2(w+1) \cdot (1+(\Delta-1)^{k-1})$.

Observe that the degree of G' is at most $\max\{\Delta+4, w+4\}$. But then $\mathrm{tw}(G') \leqslant (w+1) \cdot (1+(\Delta-1)^{k-1}) \cdot \max\{\Delta+4, w+4\}$. Since G is a minor of G', this implies that $\mathrm{tw}(G) \leqslant (w+1) \cdot (1+(\Delta-1)^{k-1}) \cdot \max\{\Delta+4, w+4\}$. □

Using the same arguments as in Theorem 3.3, we can prove the following.

Theorem 3.5. *For fixed k, w, the k-Spanner of Treewidth w problem can be decided in linear time on graphs of bounded degree.*

It can be observed that our algorithmic result is tight: the Tree k-Spanner problem becomes NP-complete for $k \geqslant 4$ on graphs with at most one vertex of unbounded degree. Recall that Cai and Corneil [3] proved that, for a given graph G, the problem to decide whether G has a tree k-spanner is NP-complete for any fixed $k \geqslant 4$. The reduction of Cai and Corneil is from the well-known NP-complete 3-Satisfiability problem. Kratochvíl [14] showed the NP-completeness of 3-Satisfiability with the additional restriction that each clause has two or three literals, and each variable occurs exactly once in positive and exactly two times in negation. It is sufficient to use this variant of 3-Satisfiability in the construction proposed by Cai and Corneil to prove the following.

Proposition 3.6. *For any fixed $k \geqslant 4$, it is* NP-*complete to decide whether a graph G with at most one vertex of degree greater than* 8 *has a tree k-spanner.*

4. Conclusions

We showed that the Tree k-Spanner problem and the k-Spanner of Treewidth w problem are fixed-parameter tractable on graphs of bounded degree. It would be interesting to show that one could use tools that do not rely on Courcelle's theorem or Bodlaender's algorithm to speed up practical implementations.

References

[1] I. Abraham, Y. Bartal, O. Neiman, Nearly tight low stretch spanning trees, in: FOCS, IEEE Computer Society, 2008, pp. 781–790.
[2] H.L. Bodlaender, A linear-time algorithm for finding tree-decompositions of small treewidth, SIAM J. Comput. 25 (1996) 1305–1317.
[3] L. Cai, D.G. Corneil, Tree spanners, SIAM J. Discrete Math. 8 (1995) 359–387.
[4] B. Courcelle, The monadic second-order logic of graphs. I. Recognizable sets of finite graphs, Inform. and Comput. 85 (1990) 12–75.
[5] B. Courcelle, The monadic second-order logic of graphs. III: Tree-decompositions, minor and complexity issues, ITA 26 (1992) 257–286.
[6] F.F. Dragan, F.V. Fomin, P.A. Golovach, A PTAS for the sparsest spanners problem on apex-minor-free graphs, in: E. Ochmanski, J. Tyszkiewicz (Eds.), MFCS, in: Lecture Notes in Computer Science, vol. 5162, Springer, 2008, pp. 290–298.
[7] M. Elkin, Y. Emek, D.A. Spielman, S.-H. Teng, Lower-stretch spanning trees, SIAM J. Comput. 38 (2008) 608–628.
[8] Y. Emek, D. Peleg, Approximating minimum max-stretch spanning trees on unweighted graphs, in: J.I. Munro (Ed.), SODA, SIAM, 2004, pp. 261–270.
[9] J. Fakcharoenphol, S. Rao, K. Talwar, A tight bound on approximating arbitrary metrics by tree metrics, in: STOC, ACM, 2003, pp. 448–455.
[10] S.P. Fekete, J. Kremer, Tree spanners in planar graphs, Discrete Appl. Math. 108 (2001) 85–103.
[11] J. Flum, M. Grohe, Parameterized Complexity Theory, Texts in Theoretical Computer Science. An EATCS Series, Springer-Verlag, Berlin, 2006.
[12] M. Herlihy, S. Tirthapura, R. Wattenhofer, Competitive concurrent distributed queuing, in: PODC, 2001, pp. 127–133.
[13] T. Kloks, Treewidth. Computations and Approximations, Lecture Notes in Computer Science, vol. 842, Springer, Berlin, 1994.
[14] J. Kratochvíl, A special planar satisfiability problem and a consequence of its NP-completeness, Discrete Appl. Math. 52 (1994) 233–252.
[15] R. Niedermeier, Invitation to fixed-parameter algorithms, Oxford Lecture Series in Mathematics and its Applications, vol. 31, Oxford University Press, Oxford, 2006.
[16] M.I. Ostrovskii, Minimal congestion trees, Discrete Math. 285 (2004) 219–226.
[17] Y. Otachi, H.L. Bodlaender, E.J. van Leeuwen, Complexity results for the spanning tree congestion problem, in: D.M. Thilikos (Ed.), WG, in: Lecture Notes in Computer Science, vol. 6410, Springer, 2010, pp. 3–14, doi:10.1007/978-3-642-16926-7_3.
[18] D. Peleg, Low stretch spanning trees, in: K. Diks, W. Rytter (Eds.), MFCS, in: Lecture Notes in Computer Science, vol. 2420, Springer, 2002, pp. 68–80.
[19] D. Peleg, A.A. Schäffer, Graph spanners, J. Graph Theory 13 (1989) 99–116.
[20] D. Peleg, J.D. Ullman, An optimal synchronizer for the hypercube, SIAM J. Comput. 18 (1989) 740–747.