Information Processing Letters 154 (2020) 105873

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

An eccentricity 2-approximating spanning tree of a chordal graph is computable in linear time

Feodor F. Dragan

Algorithmic Research Laboratory, Department of Computer Science, Kent State University, Kent, OH, USA

ARTICLE INFO

Article history:
Received 6 October 2018
Received in revised form 9 October 2019
Accepted 13 October 2019
Available online 24 October 2019
Communicated by Ryuhei Uehara

Keywords:
Linear-time algorithm
Chordal graphs
Eccentricities
Approximation
Spanning trees

ABSTRACT

It is known that every chordal graph $G = (V, E)$ has a spanning tree T such that, for every vertex $v \in V$, $ecc_T(v) \le ecc_G(v) + 2$ holds (here $ecc_G(v) := \max\{d_G(v, u) : u \in V\}$ is the eccentricity of v in G). We show that such a spanning tree can be computed in linear time for every chordal graph. As a byproduct, we get that the eccentricities of all vertices of a chordal graph G can be computed in linear time with an additive one-sided error of at most 2, i.e., after a linear time preprocessing, for every vertex v of G, one can compute in $O(1)$ time an estimate $\hat{e}(v)$ of its eccentricity $ecc_G(v)$ such that $ecc_G(v) \le \hat{e}(v) \le ecc_G(v) + 2$.

Introduction. All graphs $G = (V, E)$ in this note are connected, finite, unweighted, undirected, loopless and without multiple edges. The *length of a path* from a vertex v to a vertex u is the number of edges in the path. The *distance* $d_G(u, v)$ between vertices u and v is the length of a shortest path connecting u and v in G. The *eccentricity* of a vertex v, denoted by $ecc_G(v)$, is the largest distance from that vertex v to any other vertex, i.e., $ecc_G(v) = \max_{u \in V} d_G(v, u)$. A graph G is called *chordal* if all its induced cycles have length 3.

Eccentricity k-approximating spanning trees were introduced by Prisner in [12]. A spanning tree T of a graph G is called an *eccentricity k-approximating spanning tree* if for every vertex v of G, $ecc_T(v) \le ecc_G(v) + k$ holds [12]. Prisner demonstrated in [12], that every chordal graph has an eccentricity 2-approximating spanning tree and that the bound 2 is sharp. Later this result was extended in [7] to a larger family of graphs which includes among others all chordal graphs. Any such graph admits an eccentricity 2-approximating spanning tree. Unfortunately, both papers need $O(nm)$ time to construct such a spanning tree for an n-vertex, m-edge chordal graph, making this a more existential-type result than a result useful for efficient approximation of all eccentricities. In fact, in $O(nm)$ time, all *exact* vertex eccentricities can be computed in any graph. Moreover, a recent paper [9] demonstrated that in any graph an eccentricity k-approximating spanning tree with minimum k can be found in $O(nm)$ time.

In this note, using two ingredients known from literature and one new ingredient, we show that an eccentricity 2-approximating spanning tree of any chordal graph can be computed in linear time. This allows computation of eccentricities of all vertices of a chordal graph G with an additive one-sided error of at most 2 in total linear time. In particular, we get that after a linear time preprocessing, for every vertex v of G, one can compute in $O(1)$ time an estimate $\hat{e}(v)$ of its eccentricity $ecc_G(v)$ such that $ecc_G(v) \le \hat{e}(v) \le ecc_G(v) + 2$.

Recently, in [4], it was shown that every graph with δ-thin geodesic triangles admits an eccentricity (2δ)-approximating spanning tree constructible in $O(\delta|E|)$ time. As in chordal graphs all geodesic triangles are 2-thin [4], an immediate consequence of that result is that an eccen-

E-mail address: dragan@cs.kent.edu.

https://doi.org/10.1016/j.ipl.2019.105873

tricity 4-approximating spanning tree of a chordal graph is constructible in linear time. Here, we improve the error from 4 to optimal 2.

In what follows we will need a few more notions and notations. The *radius* $rad(G)$ of a graph G is the minimum eccentricity of a vertex in G, i.e., $rad(G) = \min_{v\in V} \max_{u\in V} d_G(v,u)$. A vertex c with $ecc_G(c) = rad(G)$ is called a central vertex of G. The *center* $C(G) = \{c \in V : ecc_G(c) = rad(G)\}$ of a graph G is the set of all its central vertices. The *diameter* $diam(G)$ of a graph G is the largest distance between a pair of vertices in G, i.e., $diam(G) = \max_{u,v\in V} d_G(u,v) = \max_{v\in V} ecc_G(v)$. A pair of vertices u, v of G with $diam(G) = d_G(u,v)$ is called a *diametral pair* and any shortest path between u and v is called a *diametral path* of G. Two vertices u, v of G are called *mutually distant vertices* if $d_G(u,v) = ecc_G(v) = ecc_G(u)$. Denote also by $F(v) = \{u \in V : d_G(v,u) = ecc_G(v)\}$ the set of all vertices of G that are *most distant* from v. For a vertex $v \in V$ and a subset $S \subseteq V$, let $d_G(v,S) = \min\{d_G(v,u) : u \in S\}$. Furthermore, for a vertex v and a path P of G, denote by $d_G(v,P)$ the distance between v and a closest to v vertex from P.

The *disk* $D_r(s)$ of a graph G centered at vertex $s \in V$ and with radius r is the set of all vertices with distance at most r from s (i.e., $D_r(s) = \{v \in V : d_G(v,s) \le r\}$). For any two vertices u, v of G, $I(u,v) = \{z \in V : d(u,v) = d(u,z) + d(z,v)\}$ is the (metric) *interval* between u and v, i.e., all vertices that lay on shortest paths between u and v. The set $S_k(x,y) = \{z \in I(x,y) : d(z,x) = k\}$ is called a *slice* of the interval from x to y. Denote by $P(x,y) = (x = v_0, v_1, \ldots, v_{k-1}, v_k = y)$ a path connecting vertices x and y.

Previously known facts. A linear time algorithm for finding a central vertex of an arbitrary chordal graph G that was presented in [3] is crucial to our linear time algorithm for constructing an eccentricity 2-approximating spanning tree for G. It was shown [3] that for every vertex s of a chordal graph G, every vertex $z \in F(s)$ has the eccentricity at least $\max\{2rad(G) - 3, diam(G) - 2\}$, and the bound is sharp. Hence, z and $v \in F(z)$ or v and $u \in F(v)$ or u and $w \in F(u)$ are mutually distant vertices. The algorithm of [3] starts with finding in linear time such a pair x, y of mutually distant vertices. Then, it carefully picks in linear time a special vertex c in a middle slice $S_{\lfloor d(x,y)/2 \rfloor}(x,y)$ of the interval $I(x,y)$. Finally, if c is not a central vertex of G, then [3] shows that the eccentricity of any vertex $t \in F(c)$ is larger than $d_G(x,y)$, and the process can be started again with a new improved pair of mutually distant vertices. Since there can only be at most two improvements on the initial distance $d_G(x,y)$ (from $diam(G) - 2$ to $diam(G) - 1$ and from $diam(G) - 1$ to $diam(G)$), the whole algorithm works in linear time. As a byproduct of this algorithm, we can claim the following additional property of the central vertex found by the algorithm of [3].

Fact 1 *([3]). A central vertex of a chordal graph that is also a middle vertex of a shortest path of length at least* $\max\{2rad(G) - 3, diam(G) - 2\}$ *can be found in linear time.*

By a later result in [5,8], the number of improvements on the initial distance $d_G(x,y)$ in the algorithm of [3] can be reduced by one if, instead of any furthest vertex from s, the vertex z last visited by a $LBFS(s)$ is used. A *Lexicographic-Breadth-First-Search*, $LBFS(s)$, starting at vertex s is a refined variant of a *Breadth-First-Search*, $BFS(s)$, with a strict tie-breaking rule (see [14]). It still runs in linear time for any graph [11].

Fact 2 *([5,8]). Let z be the vertex of a chordal graph G last visited by a $LBFS$. Then, $ecc_G(z) \ge diam(G) - 1$. Furthermore, if $diam(G)$ is even or $ecc_G(z)$ is odd then $ecc_G(z) = diam(G)$.*

This strong fact may seem to suggest that the diameter of a chordal graph might be computable in linear time as well. However, that is very unlikely as an algorithm that can distinguish between diameter 2 and 3 in a sparse chordal graph in subquadratic time will refute the widely believed *Orthogonal Vectors Conjecture* (see [5,13]).

Since for any chordal graph G, $diam(G) \ge 2rad(G) - 2$ holds [1,2], from Fact 2 we get that $ecc_G(z)$ is not the diameter $diam(G)$ only if $diam(G) = 2rad(G) - 1 = ecc_G(z) + 1$. Note that, for any graph G, $2rad(G) \ge diam(G)$ holds. Thus, regardless of $ecc_G(z)$ is $diam(G)$ or not, $ecc_G(z) \ge 2rad(G) - 2$ must hold. Thus, we have the following slight improvement of Fact 1, which will be handy later.

Fact 3. *A central vertex of a chordal graph that is also a middle vertex of a shortest path P of length at least* $\max\{2rad(G) - 2, diam(G) - 1\}$ *can be found in linear time. Furthermore, if $diam(G)$ is even or the length of P is odd, then P is a diametral path of G.*

Fact 3 is the first ingredient to our main result. The second ingredient is a nice property of the eccentricity function in chordal graphs established in [7] (even for a larger family of graphs).

Fact 4 *([7]). For every chordal graph G and any its vertex v, the following formula is true:*

$$d_G(v, C(G)) + rad(G) - \epsilon \le ecc_G(v) \le d_G(v, C(G)) + rad(G),$$

where $\epsilon \le 1$, if $diam(G) = 2rad(G)$, and $\epsilon = 0$, otherwise.

We will need also the following auxiliary lemma.

Lemma 1 *([6,10]). If vertices a and b of a disk $D_r(u)$ of a chordal graph are connected by a path $P(a,b)$ outside of $D_r(u)$ [i.e., $P(a,b) \cap D_r(u) = \{a,b\}$], then a and b must be adjacent. In particular, for every integer k and every pair of vertices x and y, slice $S_k(x,y)$ forms a clique.*

One more ingredient and the main result. Our third ingredient is that, in a chordal graph G, a middle vertex of a shortest path of length at least $2rad(G) - 2$ is within distance at most two from every central vertex of G.

Fact 5. *Let G be a chordal graph and c be a middle vertex of a shortest path P of length at least $2rad(G) - 2$ in G. Then, $C(G) \subseteq D_2(c)$. Furthermore, if the length of P is $2rad(G)$ then $C(G) \subseteq D_1(c)$.*

Proof. Let $P(x, y)$ be a shortest path between vertices x and y, $d_G(x, y) \geq 2rad(G) - 2$, and c be the vertex of $P(x, y)$ at distance $\lfloor d(x, y)/2 \rfloor$ from x. Consider an arbitrary vertex $v \in C(G)$. We know that both $d_G(v, x)$ and $d_G(v, y)$ are at most $rad(G)$. Consider arbitrary shortest paths $P(v, x)$ and $P(v, y)$ and denote by $P(y, c)$ the subpath of $P(x, y)$ between y and c.

If $d_G(x, y) = 2rad(G)$, then both c and v are in $S_{rad(G)}(x, y)$ and, by Lemma 1, $d_G(c, v) \leq 1$.

Assume now that $d_G(x, y) = 2rad(G) - 1$. Then $d_G(x, c) = rad(G) - 1$ and $d_G(y, c) = rad(G)$. If also $d_G(x, v) = rad(G) - 1$, then both c and v are in $S_{rad(G)-1}(x, y)$ and, by Lemma 1, $d_G(c, v) \leq 1$. So, let $d_G(x, v) = rad(G)$, and consider the vertex t on path $P(x, v)$ adjacent to v. Vertices t and c belong to $D_{rad(G)-1}(x)$ and are connected by a path $\{t\} \cup P(v, y) \setminus \{y\} \cup P(y, c)$ outside of $D_{rad(G)-1}(x)$ (note that $d_G(x, P(v, y)) \geq rad(G)$ as $d_G(v, y) \leq rad(G)$ and $d_G(x, y) = 2rad(G) - 1$). By Lemma 1, $d_G(c, t) \leq 1$ and hence $d_G(c, v) \leq 2$.

Finally, assume that $d_G(x, y) = 2rad(G) - 2$. Then $d_G(x, c) = rad(G) - 1 = d_G(y, c)$. If $d_G(x, v) \leq rad(G) - 1$ and $d_G(y, v) \leq rad(G) - 1$, then both c and v are in $S_{rad(G)-1}(x, y)$ and, by Lemma 1, $d_G(c, v) \leq 1$. So, without loss of generality, let $d_G(x, v) = rad(G)$. Consider the vertex t on path $P(x, v)$ adjacent to v. If $d_G(x, P(v, y)) \geq rad(G)$, then as before we get $d_G(c, t) \leq 1$ and hence $d_G(c, v) \leq 2$ (since vertices t and c belong to $D_{rad(G)-1}(x)$ and are connected by a path $\{t\} \cup P(v, y) \setminus \{y\} \cup P(y, c)$ outside of $D_{rad(G)-1}(x)$). If now $d_G(x, P(v, y)) \leq rad(G) - 1$, then to keep $d_G(x, y) = 2rad(G) - 2$, only the neighbor s of v on shortest path $P(v, y)$ can be at distance $rad(G) - 1$ from x (all other vertices of $P(v, y)$ must be at distance at least $rad(G)$ from x). Necessarily, $d_G(s, y) = rad(G) - 1$. But now, both s and c belong to $S_{rad(G)-1}(x, y)$. By Lemma 1, $d_G(c, s) \leq 1$ and hence $d_G(c, v) \leq 2$. □

We are ready to prove our main result.

Theorem 1. *An eccentricity 2-approximating spanning tree of a chordal graph G can be computed in linear time.*

Proof. By Fact 3, a central vertex c of a chordal graph that is also a middle vertex of a shortest path P of length at least $2rad(G) - 2$ can be found in linear time. Furthermore, if $diam(G) = 2rad(G)$ then P is a diametral path of G. By Fact 5, $C(G) \subseteq D_2(c)$, and even $C(G) \subseteq D_1(c)$ if $diam(G) = 2rad(G)$. We can show now that any shortest path tree T of G rooted at c is an eccentricity 2-approximating spanning tree of G.

Consider an arbitrary vertex v in G and let v' be a vertex of $C(G)$ closets to v. By Fact 4, $ecc_G(v) \geq d_G(v, C(G)) + rad(G) - \epsilon = d_G(v, v') + rad(G) - \epsilon$, where $\epsilon \leq 1$, if $diam(G) = 2rad(G)$, and $\epsilon = 0$, otherwise. Since T is a shortest path tree and c is a central vertex of G, $ecc_T(v) \leq d_T(v, c) + ecc_T(c) = d_G(v, c) + ecc_G(c) = d_G(v, c) + rad(G)$. Hence, by the triangle inequality,

$$\begin{aligned} ecc_T(v) - ecc_G(v) &\leq d_G(v, c) + rad(G) - d_G(v, v') \\ &\quad - rad(G) + \epsilon \qquad (1) \\ &\leq d_G(c, v') + \epsilon \\ &\leq 2. \end{aligned}$$

Recall that if $diam(G) < 2rad(G)$ then $d_G(c, v') \leq 2$ and $\epsilon = 0$, and if $diam(G) = 2rad(G)$ then $d_G(c, v') \leq 1$ and $\epsilon \leq 1$. □

Note that the eccentricities of all vertices in any tree $T = (V, U)$ can be computed in $O(|V|)$ total time. It is a folklore by now that for trees the following facts are true:

(1) The center $C(T)$ of any tree T consists of one vertex or two adjacent vertices.
(2) The center $C(T)$ and the radius $rad(T)$ of any tree T can be found in linear time.
(3) For every vertex $v \in V$, $ecc_T(v) = d_T(v, C(T)) + rad(T)$.

Hence, using $BFS(C(T))$ on T one can compute $d_T(v, C(T))$ for all $v \in V$ in total $O(|V|)$ time. Adding now $rad(T)$ to $d_T(v, C(T))$, one gets $ecc_T(v)$ for all $v \in V$. Consequently, by Theorem 1, we get the following additive approximations for the vertex eccentricities in chordal graphs.

Corollary 1. *Let $G = (V, E)$ be a chordal graph. There is an algorithm which in total linear ($O(|E|)$) time outputs for every vertex $v \in V$ an estimate $\hat{e}(v)$ of its eccentricity $ecc_G(v)$ such that $ecc_G(v) \leq \hat{e}(v) \leq ecc_G(v) + 2$.*

Concluding remark. We demonstrated that an eccentricity 2-approximating spanning tree of a chordal graph can be computed in linear time. Can this result be extended to a more general class of graphs described in [7] (they all admit eccentricity 2-approximating spanning trees). The main bottleneck there is whether a central vertex in such a graph can be found in linear time. It is interesting also whether a linear time algorithm exists which for every chordal graph G computes estimates $\hat{e}(v)$, $v \in V$, with $ecc_G(v) \leq \hat{e}(v) \leq ecc_G(v) + \mu$ for $\mu \leq 1$.

Declaration of competing interest

There are no conflicts of interest.

References

[1] G.J. Chang, G.L. Nemhauser, The k-domination and k-stability problems on sun-free chordal graphs, SIAM J. Algebraic Discrete Methods 5 (1984) 332–345.
[2] V. Chepoi, Centers of triangulated graphs, Math. Notes 43 (1988) 143–151.
[3] V.D. Chepoi, F.F. Dragan, A linear-time algorithm for finding a central vertex of a chordal graph, in: Jan van Leeuwen (Ed.), "Algorithms - ESA'94" Second Annual European Symposium, Utrecht, The Netherlands, September 1994, in: LNCS, vol. 855, Springer, 1994, pp. 159–170.

[4] V. Chepoi, F.F. Dragan, M. Habib, Y. Vaxés, H. Alrasheed, Fast approximation of centrality and distances in hyperbolic graphs, CoRR, arXiv:1805.07232, 2018.

[5] D.G. Corneil, F.F. Dragan, M. Habib, C. Paul, Diameter determination on restricted graph families, Discrete Appl. Math. 113 (2001) 143–166.

[6] D.G. Corneil, F.F. Dragan, E. Köhler, On the power of BFS to determine a graph's diameter, Networks 42 (2003) 209–222.

[7] F.F. Dragan, E. Köhler, H. Alrasheed, Eccentricity approximating trees, Discrete Appl. Math. 232 (2017) 142–156.

[8] F.F. Dragan, F. Nicolai, A. Brandstädt, LexBFS-orderings and powers of graphs, in: Proceedings of the WG'96, in: Lecture Notes in Computer Science, vol. 1197, Springer, Berlin, 1997, pp. 166–180.

[9] G. Ducoffe, Easy computation of eccentricity approximating trees, Discrete Appl. Math. 260 (2019) 267–271.

[10] M. Farber, R.E. Jamison, Convexity in graphs and hypergraphs, SIAM J. Algebraic Discrete Methods 7 (1986) 433–444.

[11] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, New York, 1980.

[12] E. Prisner, Eccentricity-approximating trees in chordal graphs, Discrete Math. 220 (2000) 263–269.

[13] L. Roditty, V. Vassilevska Williams, Fast approximation algorithms for the diameter and radius of sparse graphs, in: STOC 2013, 2013, pp. 515–524.

[14] D. Rose, R.E. Tarjan, G. Lueker, Algorithmic aspects on vertex elimination on graphs, SIAM J. Comput. 5 (1976) 266–283.