Algorithms for outerplanar graph roots and graph roots of pathwidth at most 2*

Petr A. Golovach[1], Pinar Heggernes[1], Dieter Kratsch[2], Paloma T. Lima[1], and Daniël Paulusma[3]

[1] Department of Informatics, University of Bergen, PB 7803, 5020 Bergen, Norway, {petr.golovach,pinar.heggernes,paloma.lima}@uib.no
[2] Laboratoire d'Informatique Théorique et Appliquée, Université de Lorraine, 57045 Metz Cedex 01, France, dieter.kratsch@univ-lorraine.fr
[3] School of Engineering and Computing Sciences, Durham University, Durham DH1 3LE, UK, daniel.paulusma@durham.ac.uk

Abstract. Deciding whether a given graph has a square root is a classical problem that has been studied extensively both from graph theoretic and from algorithmic perspectives. The problem is NP-complete in general, and consequently substantial effort has been dedicated to deciding whether a given graph has a square root that belongs to a particular graph class. There are both polynomial-time solvable and NP-complete cases, depending on the graph class. We contribute with new results in this direction. Given an arbitrary input graph G, we give polynomial-time algorithms to decide whether G has an outerplanar square root, and whether G has a square root that is of pathwidth at most 2.

1 Introduction

Squares and square roots of graphs form a classical and well-studied topic in graph theory, which has also attracted significant attention from the algorithms community. A graph G is the *square* of a graph H if G and H have the same vertex set, and two vertices are adjacent in G if and only if the distance between them is at most 2 in H. This situation is denoted by $G = H^2$, and H is called a *square root* of G. A square root of a graph need not be unique; it might even not exist. There are graphs without square roots, graphs with a unique square root, and graphs with several different square roots. Characterizing and recognizing graphs with square roots has therefore been an intriguing and important problem both in graph theory and in algorithms for decades.

Already in 1967, Mukhopadhyay [26] proved that a graph G on vertex set $\{v_1, \dots, v_n\}$ has a square root if and only if G contains complete subgraphs $\{K^1, \dots, K^n\}$, such that each K^i contains v_i, and vertex v_j belongs to K^i if and only if v_i belongs to K^j. Unfortunately, this characterization does not yield a polynomial-time algorithm for deciding whether G has a square root. Let us formally call Square Root the problem of deciding whether an input graph G has

* Supported by the Research Council of Norway via the project "CLASSIS" and the Leverhulme Trust (RPG-2016-258).

a square root. In 1994, it was shown by Motwani and Sudan [25] that Square Root is NP-complete. Motivated by its computational hardness, special cases of the problem have been studied, where the input graph G belongs to a particular graph class. According to these results, Square Root is polynomial-time solvable on planar graphs [22], and more generally, on every non-trivial minor-closed graph class [27]. Polynomial-time algorithms exist also when the input graph G belongs to one of the following graph classes: block graphs [20], line graphs [23], trivially perfect graphs [24], threshold graphs [24], graphs of maximum degree 6 [3], graphs of maximum average degree smaller than $\frac{46}{11}$ [13], graphs with clique number at most 3 [14], and graphs with bounded clique number and no long induced path [14]. On the negative side, it has been shown that Square Root is NP-complete on chordal graphs [17]. A number of parameterized complexity results exist for the problem [3, 4, 13].

More interesting from our perspective, the intractability of the problem has also been attacked by restricting the properties of the square root that we are looking for. In this case, the input graph G is arbitrary, and the question is whether G has a square root that belongs to some graph class $\mathcal{H}$ specified in advance. We denote this problem by $\mathcal{H}$-Square Root, and this is exactly the problem variant that we focus on in this paper.

Significant advances have been made also in this direction. Previous results show that $\mathcal{H}$-Square Root is polynomial-time solvable for the following graph classes $\mathcal{H}$: trees [22], proper interval graphs [17], bipartite graphs [16], block graphs [20], strongly chordal split graphs [21], ptolemaic graphs [18], 3-sun-free split graphs [18], cactus graphs [12], cactus block graphs [8] and graphs with girth at least g for any fixed $g \geq 6$ [10]. The result for 3-sun-free split graphs has been extended to a number of other subclasses of split graphs in [19]. Observe that if $\mathcal{H}$-Square Root is polynomial-time solvable for some class $\mathcal{H}$, then this does not automatically imply that $\mathcal{H}'$-Square Root is polynomial-time solvable for a subclass $\mathcal{H}'$ of $\mathcal{H}$.

On the negative side, $\mathcal{H}$-Square Root remains NP-complete for each of the following graph classes $\mathcal{H}$: graphs of girth at least 5 [9], graphs of girth at least 4 [10], split graphs [17], and chordal graphs [17]. All known NP-hardness constructions involve dense graphs [9, 10, 17, 25], and the square roots that occur in these constructions are dense as well. This, in combination with the listed polynomial-time cases, naturally leads to the question whether $\mathcal{H}$-Square Root is polynomial-time solvable if the class $\mathcal{H}$ is "sparse" in some sense.

Motivated by the above, in this paper we study $\mathcal{H}$-Square Root when $\mathcal{H}$ is the class of outerplanar graphs, and when $\mathcal{H}$ is the class of graphs of pathwidth at most 2. In both cases, we show that $\mathcal{H}$-Square Root can be solved in polynomial time. In particular, we prove that Outerplanar (Square) Root can be solved in time $O(n^4)$ and (Square) Root of Pathwidth ≤ 2 in time $O(n^6)$. Our approach for outerplanar graphs can in fact be directly applied to every subclass of outerplanar graphs that is closed under edge deletion and that can be expressed in monadic second-order logic, including cactus graphs, for which a polynomial-time algorithm is already known [12]. Due to space restrictions, some proofs are omitted; see [11] for the full version of our paper.

2 Preliminaries

We consider only finite undirected graphs without loops and multiple edges. We refer to the textbook by Diestel [7] for any undefined graph terminology.

Let G be a graph. We denote the vertex set of G by V_G and the edge set by E_G. The subgraph of G induced by a subset $U \subseteq V_G$ is denoted by $G[U]$. The graph $G - U$ is the graph obtained from G after removing the vertices of U. If $U = \{u\}$, we also write $G - u$. Similarly, we denote the graph obtained from G by deleting a set of edges S, or a single edge e, by $G - S$ and $G - e$, respectively.

The *distance* $\mathrm{dist}_G(u, v)$ between a pair of vertices u and v of G is the number of edges of a shortest path between them. The *open neighborhood* of a vertex $u \in V_G$ is defined as $N_G(u) = \{v \mid uv \in E_G\}$, and its *closed neighborhood* is defined as $N_G[u] = N_G(u) \cup \{u\}$. For $S \subseteq V_G$, $N_G(S) = (\bigcup_{v \in S} N_G(v)) \setminus S$. Two (adjacent) vertices u, v are said to be *true twins* if $N_G[u] = N_G[v]$. A vertex v is *simplicial* if $N_G[v]$ is a clique, that is, if there is an edge between any two vertices of $N_G[v]$. The *degree* of a vertex $u \in V_G$ is defined as $d_G(u) = |N_G(u)|$. The maximum degree of G is $\Delta(G) = \max\{d_G(v) \mid v \in V_G\}$. A vertex of degree 1 is said to be a *pendant* vertex.

A *connected component* of G is a maximal connected subgraph. A vertex u is a *cut vertex* of a graph G with at least two vertices if $G - u$ has more components than G. A connected graph without cut vertices is said to be *biconnected*. An inclusion-maximal induced biconnected subgraph of G is called a *block*.

For a positive integer k, the *k-th power* of a graph H is the graph $G = H^k$ with vertex set $V_G = V_H$ such that every pair of distinct vertices u and v of G are adjacent if and only if $\mathrm{dist}_H(u, v) \leq k$. For the particular case $k = 2$, H^2 is a *square* of H, and H is a *square root* of G if $G = H^2$.

The *contraction* of an edge uv of a graph G is the operation that deletes the vertices u and v and replaces them by a vertex w adjacent to $(N_G(u) \cup N_G(v)) \setminus \{u, v\}$. A graph G' is a contraction of a graph G if G' can be obtained from G by a series of edge contraction. A graph G' is a *minor* of G if it can be obtained from G by vertex deletions, edge deletions and edge contractions.

A graph G is *planar* if it admits an embedding on the plane such that there are no edges crossing (except in endpoints). A planar graph G is *outerplanar* if it admits a crossing-free embedding on the plane in such a way that all its vertices are on the boundary of the same (external) face. For a considered outerplanar graph, we always assume that its embedding on the plane is given. If G is a planar biconnected graph different from K_2, then for any of its embeddings, the boundary of each face is a cycle (see, e.g., [7]). If G is a biconnected outerplanar graph distinct from K_2, then the cycle C forming the boundary of the external face is unique and we call it the *boundary cycle*. By definition, all vertices of G are laying on C, and every edge is either an edge of C or a *chord* of C, that is, its endpoints are vertices of C that are non-adjacent in C. Clearly, these chords are not intersecting in the embedding. For a vertex u, we define the *clockwise ordering with respect to* u as a clockwise ordering of the vertices on C starting from u. For a subset of vertices X, the *clockwise ordering of* X *with respect to* u is the ordering induced by the clockwise ordering of the vertices of C. See Figure 1 a) for some examples. In our paper, we use these terms for blocks of an outerplanar graph that are distinct from K_2. Outerplanar graphs can also be characterized via forbidden minors as shown by Sysło [29].

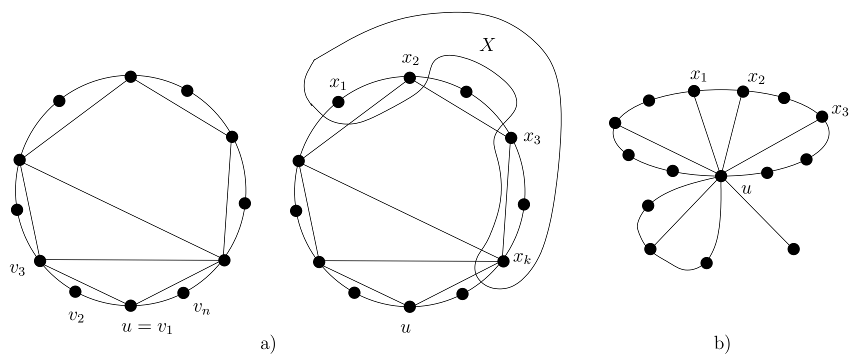

Fig. 1. a) clockwise orderings with respect to u of a biconnected outerplanar graph with vertex set $V_G = \{v_1, \ldots, v_n\}$ and a set $X = \{x_1, \ldots, x_k\}$. **b)** Example of a set $X = \{x_1, x_2, x_3\}$ that is consecutive with respect to u; notice that the vertices x_1 and x_3 are not consecutive.

Lemma 1 ([29]). *A graph* G *is outerplanar if and only if it does not contain* K_4 *and* $K_{2,3}$ *as minors.*

A *tree decomposition* of a graph G is a pair (T, X) where T is a tree and $X = \{X_i \mid i \in V_T\}$ is a collection of subsets (called *bags*) of V_G such that the following three conditions hold:

i) $\bigcup_{i \in V_T} X_i = V_G$,
ii) for each edge $xy \in E_G$, $x, y \in X_i$ for some $i \in V_T$, and
iii) for each $x \in V_G$ the set $\{i \mid x \in X_i\}$ induces a connected subtree of T.

The *width* of a tree decomposition $(\{X_i \mid i \in V_T\}, T)$ is $\max_{i \in V_T} \{|X_i| - 1\}$. The *treewidth* $\mathbf{tw}(G)$ of a graph G is the minimum width over all tree decompositions of G. If T is restricted to be a path, then we say that (X, T) is a *path decomposition* of G. The *pathwidth* $\mathbf{pw}(G)$ of G is the minimum width over all path decompositions of G. Notice that a path decomposition of G can be seen as a sequence $(X_1, \ldots, X_r)$ of bags. We always assume that the bags $(X_1, \ldots, X_r)$ are distinct and inclusion incomparable, that is, there are no bags X_i and X_j such that $X_i \subset X_j$. The following fundamental results are due to Bodlaender [1], and Bodlaender and Kloks [2].

Lemma 2 ([1, 2]). *For every fixed constant c, it is possible to decide in linear time whether the treewidth or the pathwidth of a graph is at most c.*

We need the following three folklore observations about treewidth.

Observation 1 *If H is a minor (contraction) of G, then $\mathbf{tw}(H) \le \mathbf{tw}(G)$ and $\mathbf{pw}(H) \le \mathbf{pw}(G)$.*

Observation 2 *For an outerplanar graph G, $\mathbf{tw}(G) \le 2$.*

Observation 3 *For a graph G and a positive integer k,*

$$\mathbf{tw}(G^k) \le (\mathbf{tw}(G) + 1)\Delta(G)^{\lfloor k/2 \rfloor + 1}$$

and

$$\mathbf{pw}(G^k) \le (\mathbf{pw}(G) + 1)\Delta(G)^{\lfloor k/2 \rfloor + 1}.$$

Let H be a square root of a graph G. We say that H is a *minimal* square root of G if $H^2 = G$, and no proper subgraph of H is a square root of G. We need the following simple observations.

Observation 4 *Let $\mathcal{H}$ be a graph class closed under edge deletion. If a graph G has a square root $H \in \mathcal{H}$, then G has a minimal square root that belongs to $\mathcal{H}$.*

Observation 5 *Let H be a minimal square root of a graph G that contains three vertices u, v, w that are pairwise adjacent in H. Then v or w has a neighbor $x \neq u$ in H such that x is not adjacent to u in H.*

We conclude this section by a lemma that is implicit in [12], which enables us to identify some edges that are not included in any square root.

Lemma 3. *Let x, y be distinct neighbors of a vertex u in a graph G such that x and y are at distance at least 3 in $G - u$. Then $xu, yu \notin E_H$ for any square root H of G.*

3 Outerplanar Roots

In this section, we show that it can be decided in polynomial time whether a graph has an outerplanar square root. We say that a square root H of G is an *outerplanar root* if H is outerplanar. We define the following problem:

OUTERPLANAR ROOT
Input: a graph G.
Question: is there an outerplanar graph H such that $H^2 = G$?

The main result of this section is the following.

Theorem 1. OUTERPLANAR ROOT *can be solved in time $O(n^4)$, where n is the number of vertices of the input graph.*

The remaining part of this section is devoted to the proof of Theorem 1. In Section 3.1 we obtain several structural results we need to construct an algorithm for OUTERPLANAR ROOT. Then, in Section 3.2, we construct a polynomial-time algorithm for OUTERPLANAR ROOT.

3.1 Structural Lemmas

In this section, we give several structural results about outerplanar square roots. Due to space restriction we omit the proofs.

Let H be an outerplanar root of a graph G and let $u \in V_G$. We say that two distinct vertices $x, y \in N_H(u)$ are *consecutive with respect to* u if x and y are in the same block F of H and there are no vertices of $N_H(u)$ between x and y in the clockwise ordering of the vertices of the boundary cycle of F with respect to u. For a set of vertices $X \subseteq N_H(u)$, we say that the vertices X are *consecutive with respect to* u if the vertices of X are in the same block of H and any two vertices of X consecutive in the clockwise ordering of elements of X with respect to u are consecutive with respect to u; a single-vertex set is assumed to be consecutive (see Figure 1 b) for an example).

As every subgraph of an outerplanar graph is outerplanar, by Observation 4, we may restrict ourselves to minimal outerplanar roots. Let H be a minimal outerplanar root of a graph G and let $u \in V_G$. Denote by $S(G, H, u)$ a collection of all subsets X of $N_H(u)$ such that $X = N_G(x) \cap N_H(u)$ for some $x \in N_G(u) \setminus N_H(u)$. We can use $S(G, H, u)$ to find edges with both endpoints in $N_H(u)$ that are not included in a square root.

Lemma 4. *Let H be a minimal outerplanar root of a graph G, and let $u \in V_G$. Then for each $X \in S(G, H, u)$, X is consecutive with respect to u.*

Lemma 5. *Let H be a minimal outerplanar root of a graph G, and let $u \in V_G$. If for two distinct vertices $x, y \in N_H(u)$ there is no set $X \in S(G, H, u)$ such that $x, y \in X$, then $xy \notin E_H$.*

We also need the following two lemmas.

Lemma 6. *Let H be a minimal outerplanar root of a graph G, and let $u \in V_G$. If $x \in N_H(u)$ is not a pendant vertex of H, then there is a vertex $y \in N_G(u) \setminus N_H(u)$ that is adjacent to x in G.*

Lemma 7. *Let H be a minimal outerplanar root of a graph G, and let $u \in V_G$. Then any $X \in S(G, H, u)$ has size at most 4.*

By combining Lemmas 4 and 7 we obtain the following lemma.

Lemma 8. *Let H be a minimal outerplanar root of a graph G, and let $u \in V_G$. Then the following holds.*

(i) If $x, y \in N_H(u)$ do not belong to the same block of H, then for any $X \in S(G, H, u)$, $x \notin X$ or $y \notin X$.
(ii) If F is a block of H containing u and vertices $x_1, \dots, x_k \in N_H(u)$ ordered in the clockwise order with respect to u in the boundary cycle of F, then for any $X \in S(G, H, u)$, $x_i \notin X$ or $x_j \notin X$ if $i, j \in \{1, \dots, k\}$ and $|i - j| \geq 4$.

We now state some structural results that help to decide whether an edge incident to a vertex is in an outerplanar root or not. Suppose that u and v are pendant vertices of a square root H of G and that u and v are adjacent to the same vertex of $H - \{u, v\}$. Then, in G, u and v are simplicial vertices and true twins. We use this observation in the proof of the following lemma that allows to find some pendant vertices.

Lemma 9. *Let H be a minimal outerplanar root of a graph G. If G contains at least 7 simplicial vertices that are pairwise true twins, then at least one of these vertices is a pendant vertex of H.*

We apply Lemma 3 to identify the edges incident to a vertex of sufficiently high degree in an outerplanar root using the following two lemmas.

Lemma 10. *Let G be a graph having a minimal outerplanar root H. Let also $u \in V_G$ be such that there are three distinct vertices $v_1, v_2, v_3 \in N_G(u)$ that are pairwise at distance at least 3 in $G - u$. Then for $x \in N_G(u)$, $xu \notin E_H$ if and only if there is $i \in \{1, 2, 3\}$ such that $\mathrm{dist}_{G-u}(x, v_i) \geq 3$.*

Lemma 11. *Let G be a graph having a minimal outerplanar root H such that any vertex of H has at most 7 pendant neighbors. Let also $u \in V_G$ with $d_H(u) \geq 22$. Then there are distinct $v_1, v_2, v_3 \in N_G(u)$ that are pairwise at distance at least 3 in $G - u$.*

Notice that v_1, v_2 and v_3 are in distinct components of $H - u$. We obtain that v_3 is at distance at least 3 from v_1 and v_2 in $G - u$.

The next lemma is crucial for our algorithm. To state it, we need some additional notations. Let H be a minimal outerplanar root of a graph G such that each vertex of H is adjacent to at most 7 pendant vertices. Let U be a set of vertices of H that contains all vertices of degree at least 22. For every $u \in U$ and every block F of H containing u, we do the following. Consider the set $X = N_H(u) \cap V_F$ and denote the vertices of X by $x_1, \ldots, x_k$, where these vertices are numbered in the clockwise order with respect to u. Then

- for $i, j \in \{1, \ldots, k\}$, delete the edge $x_i x_j$ from G if $|i - j| \geq 4$.
- for $i \in \{1, \ldots, k\}$, delete the edges $x_i y$ from G for $y \in N_H(u) \setminus V_F$.

Denote by $G(H, U)$ the graph obtained in the end.

Lemma 12. *There is a constant c that depends neither on G nor on H such that*

$$\mathbf{tw}(G(H, U)) \leq c.$$

3.2 The Algorithm

In this section, we construct an algorithm for Outerplanar Root with running time $O(n^4)$. Let G be the input graph. Clearly, it is sufficient to solve Outerplanar Root for connected graphs. Hence, we assume that G is connected and has $n \geq 2$ vertices.

First, we preprocess G using Lemma 9 to reduce the number of pendant vertices adjacent to the same vertex in a (potential) outerplanar root of G. To do so, we exhaustively apply the following rule.

Pendants reduction. If G has a set X of simplicial true twins of size at least 8, then delete an arbitrary $u \in X$ from G.

The following lemma shows that this rule is safe.

Lemma 13. *If $G' = G - u$ is obtained from G by the application of* **Pendant reduction***, then G has an outerplanar root if and only if G' has an outerplanar root.*

Proof. Suppose that H is a minimal outerplanar root of G. By Lemma 9, H has a pendant vertex $u \in X$. It is easy to verify that $H' = H - u$ is an outerplanar root of G'. Assume now that H' is a minimal outerplanar root of G'. By Lemma 9, H has a pendant vertex $w \in X \setminus \{u\}$, since the vertices of $X \setminus \{u\}$ are simplicial true twins of G' and $|X \setminus \{u\}| \geq 7$. Let v be the unique neighbor of w in H'. We construct H from H' by adding u and making it adjacent to v. It is readily seen that H is an outerplanar root of G. This completes the proof. $\square$

For simplicity, we call the graph obtained by exhaustive application of the pendants reduction rule G again. The following property immediately follows from the observation that any two pendant vertices of a square root H of G adjacent to the same vertex in H are true twins of G.

Lemma 14. *Every outerplanar root of G has at most* 7 *pendant vertices adjacent to the same vertex.*

In the next stage of our algorithm we label some edges of G *red* or *blue* in such a way that the edges labeled red are included in every minimal outerplanar root and the blue edges are not included in any minimal outerplanar root. We denote by R the set of red edges and by B the set of blue edges. We also construct a set of vertices U of G such that for every $u \in U$, the edges incident to u are labeled red or blue.

Labeling. Set $U = \emptyset$, $R = \emptyset$ and $B = \emptyset$. For each $u \in V_G$ such that there are three distinct vertices $v_1, v_2, v_3 \in N_G(u)$ that are at distance at least 3 from each other in $G - u$, do the following:

(i) set $U = U \cup \{u\}$,
(ii) set $B' = \{ux \in E_G \mid \text{there is } 1 \leq i \leq 3 \text{ s.t. } \mathrm{dist}_{G-u}(x, v_i) \geq 3\}$,
(iii) set $R' = \{ux \mid x \in N_G(u)\} \setminus B'$,
(iv) set $R = R \cup R'$ and $B = B \cup B'$,
(v) if $R \cap B \neq \emptyset$, then return a no-answer and stop.

Lemmas 10 and 11 imply the following statement.

Lemma 15. *If G has an outerplanar root, then* **Labeling** *does not stop in Step* (v)*, and if H is a minimal outerplanar root of G, then $R \subseteq E_H$ and $B \cap E_H = \emptyset$. Moreover, every vertex $u \in V_G$ with $d_H(u) \geq 22$ is included in U.*

Next, we find the set of edges xy with $xu, yu \in R$ for some u in R that are not included in a minimal outerplanar root.

Finding irrelevant edges. Set $S = \emptyset$. For each $u \in U$ and each pair of distinct $x, y \in N_G(u)$ such that $ux, uy \in R$ do the following.

(i) If $xy \notin E_G$, then return a no-answer and stop.
(ii) If for x and y, there is no $v \in N_G(u)$ such that $vu \in B$ and $x, y \in N_G(v)$, then include xy in S.
(iii) If $R \cap S \neq \emptyset$, then return a no-answer and stop.

Combining Lemmas 15 and 5, we obtain the following claim.

Lemma 16. *If G has an outerplanar root, then* **Finding irrelevant edges** *does not stop in Steps* (i) *and* (iii)*, and if H is a minimal outerplanar root of G, then $S \cap E_H = \emptyset$.*

Assume that we did not stop during the execution of **Finding irrelevant edges**. Let $G' = G - S$. We show the following.

Lemma 17. *The graph G has an outerplanar root if and only if there is a set $L \subseteq E_{G'}$ such that*

(i) $R \subseteq L$, $B \cap L = \emptyset$,
(ii) for any $xy \in E_{G'}$, $xy \in L$ or there is $z \in V_{G'}$ such that $xz, yz \in L$,
(iii) for any pair of distinct edges $xz, yz \in L$, $xy \in E_{G'}$ or there is $u \in U$ such that $xu, yu \in R$,
(iv) the graph $H = (V_G, L)$ is outerplanar.

Proof. Let H be a minimal outerplanar root of G. By Lemma 16, $E_H \cap S = \emptyset$, i.e., $E_H \subseteq E_{G'}$. Let $L = E_H$. It is straightforward to verify that (i)–(iv) are fulfilled. Assume now that there is $L \subseteq E_{G'}$ such that (i)–(iv) hold. Then we have that $H = (V_G, L)$ is an outerplanar root of G. ☐

To complete the description of the algorithm, it remains to show how to check the existence of a set of edges L satisfying (i)–(iv) of Lemma 17 for given G', R and B. Notice that, if G has a minimal outerplanar root H, then G' is a subgraph of the graph $G(H, U)$ constructed in Section 3.1 by Lemma 8. By Lemma 12, there is a constant c that depends neither on G nor on H such that $\mathbf{tw}(G(H, U)) \leq c$. Therefore, $\mathbf{tw}(G') \leq c$ for a yes-instance. We use Lemma 2 to verify whether it holds. If we obtain that $\mathbf{tw}(G') > c$, we conclude that we have a no-instance and stop. Otherwise, we use the celebrated theorem of Courcelle [5], which states that any problem that can be expressed in monadic second-order logic can be solved in linear time on a graph of bounded treewidth. It is straightforward to see that properties (i)–(iv) can be expressed in this logic. In particular, to express outerplanarity in (iv), we can use Lemma 1 and the well-known fact that the property that G contains F as a minor can be expressed in monadic second-order logic if F is fixed (see, e.g., the book of Courcelle and Engelfriet [6]). It immediately implies that we can decide in linear time whether

L exists or not. Notice that we can modify these arguments such that we do not only check the existence of L but also find it. To do this, we can construct a dynamic programming algorithm for graphs of bounded treewidth that finds L.

Now we evaluate the running time of our algorithm. Since it can be verified in time $O(n)$ whether two vertices of G are true twins, the classes of true twins can be constructed in time $O(n^3)$. Then we can check whether each class contains simplicial vertices in time $O(n^2)$. Therefore, **Pendant reduction** can be done in time $O(n^3)$. For every vertex u, we can compute the distances between the vertices of $N_G(u)$ in $G - u$ in time $O(n^3)$. This implies that **Labeling** can be done in time $O(n^4)$. **Finding irrelevant edges** also can be done in time $O(n^4)$ by checking $O(n^2)$ pairs of vertices x and y. Then G' can be constructed in linear time. Finally, checking whether $\mathbf{tw}(G') \leq c$ and deciding whether there is a set of edges L satisfying the required properties can be done in linear time by Lemma 2 and Courcelle's theorem [5] respectively.

Notice that we can use the same arguments to decide whether a graph G has a square root H that belongs to some subclass $\mathcal{H}$ of the class of outerplanar graphs. To be able to apply our structural lemmas, we only need the property that $\mathcal{H}$ should be closed under edge deletions. Observe also that if the properties defining $\mathcal{H}$ could be expressed in monadic second-order logic, then we can apply Courcelle's theorem [5]. It gives us the following corollary.

Corollary 1. *For every subclass $\mathcal{C}$ of the class of outerplanar graphs that is closed under edge deletions and can be expressed in monadic second-order logic, it can be decided in time $O(n^4)$ whether an n-vertex graph G has a square root $H \in \mathcal{C}$.*

4 Roots of Pathwidth at most two

Our main approach for solving Outerplanar root is general in the sense that it can be adapted to find also square roots belonging to some other graph classes. In this section, we show that there is an algorithm to decide in polynomial time whether a graph has a square root of pathwidth at most 2. Notice that graphs of pathwidth 1 are caterpillars, and that it can be decided in polynomial time whether a graph G has a square root that is a caterpillar by an easy adaptation of algorithms for finding square roots that are trees [22, 28].

We define the following problem:

Root of Pathwidth ≤ 2
Input: a graph G.
Question: is there a graph H such that $\mathbf{pw}(H) \leq 2$ and $H^2 = G$?

The main difference between our algorithm for Root of Pathwidth $\leq$ 2 and our algorithm for Outerplanar Root lies in the way properties of the involved graph classes are used. To show the structural results needed for this algorithm, we use the property that a potential square root has a path decomposition of width at most 2, instead of the existence of an outerplanar embedding used in the previous section.

We briefly sketch the proof of the following theorem.

Theorem 2. Root of Pathwidth ≤ 2 *can be solved in time* $O(n^6)$*, where* n *is the number of vertices of the input graph.*

Proof. (*Sketch.*) Let G be the input graph. It is sufficient to solve Root of Pathwidth ≤ 2 for connected graphs. Hence, we assume that G is connected and has $n \geq 2$ vertices. Notice that the class of graphs of pathwidth at most 2 is closed under edge deletions. Therefore, by Observation 4, we can consider only minimal square roots.

First, we preprocess G to reduce the number of true twins that a given vertex of V_G might have. To do so, we show that there is a constant c_1 such that if W is a set of true twins of G of size at least c_1, then for any minimal square root H of G with $\mathbf{pw}(H) \leq 2$, either W has a vertex that is pendant in H or W has distinct nonadjacent vertices x, y, z with $d_H(x) = d_H(y) = d_H(y) = 2$. It allows us to show that if G has a set of true twins W of size at least $c_1 + 1$, then by the deletion of an arbitrary $u \in W$ from G, we obtain an equivalent instance of Root of Pathwidth ≤ 2. From now on, we can assume that any set of true twins of G has size at most c_1. We need this to obtain forthcoming structural properties.

In the next stage of our algorithm, we label some edges of G *red* or *blue* in such a way that the edges labeled red are included in every minimal square root of pathwidth at most 2 and the blue edges are not included in any minimal square root of pathwidth at most 2. We denote by R the set of red edges and by B the set of blue edges. We also construct a set of vertices U of G such that for every $u \in U$, the edges incident to u are labeled red or blue.

The labeling is based on the following structural property. If there is $u \in V_G$ such that there are five distinct vertices $v_1, \ldots, v_5$ in $N_G(u)$ that are at distance at least 3 from each other in $G - u$, then for any square root H with $\mathbf{pw}(H) \leq 2$, $ux \notin E_H$ for $x \in N_G(u)$ if and only if there is $i \in \{1, \ldots, 5\}$ such that $\mathrm{dist}_{G-u}(x, v_i) \geq 3$. Respectively, if we find $u \in V_G$ with the aforementioned property that there are five distinct vertices $v_1, \ldots, v_5$ in $N_G(u)$ that are at distance at least 3 from each other in $G - u$, then we include u in U and for $x \in N_G(u)$, we label ux blue if there is $i \in \{1, \ldots, 5\}$ such that $\mathrm{dist}_{G-u}(x, v_i) \geq 3$ and we label ux red otherwise. If we get inconsistent labelings, that is, some edge should be labeled red and blue, then we stop and report that there is no square root of pathwidth at most 2.

We show that there is a constant c_2 such that, for a square root H of G with $\mathbf{pw}(H) \leq 2$, if $d_H(u) \geq c_2$, then $u \in U$ and, therefore, all the edges of G incident to u are labeled red or blue. It means that if u is a vertex of H of sufficiently high degree, then for each edge of G incident to u, we distinguish whether this edge is in a square root or not.

Next, we find the set of edges xy with $xu, yu \in R$ which for some u in U are not included in a minimal square root of pathwidth at most 2. To do it, we use Observation 5 to show that if there is no $z \in N_G(u)$ with $uz \in B$ such that $xz, yz \in E_G$, then $xy \notin E_H$ for a minimal square root H of pathwidth at most 2. Respectively, we label such edges xy blue. Again, if we get inconsistent labelings, then we stop and report that there is no square root of pathwidth at most 2.

Denote by S the set of edges labeled blue in this stage of the algorithm and let $G' = G - S$. We prove that if G has a square root of pathwidth at most 2, then there is a constant c_4 such that $\mathbf{pw}(G') \leq c_4$. The proof is based on the property that every vertex of degree at least c_2 in a (potential) square root of pathwidth at most 2 is included in U. We can verify whether $\mathbf{pw}(G') \leq c_4$ in linear time using Lemma 2. If $\mathbf{pw}(G') > c_4$, then we stop and report that there is no square root of pathwidth at most 2. Otherwise, we obtain a path decomposition of G' of width at most c_4.

Then, similarly to the proof of Theorem 1, we obtain that G has a square root of pathwidth at most 2 if and only if there is a set $L \subseteq E_{G'}$ such that

(i) $R \subseteq L$, $B \cap L = \emptyset$,
(ii) for any $xy \in E_{G'}$, $xy \in L$ or there is $z \in V_{G'}$ such that $xz, yz \in L$,
(iii) for any distinct edges $xz, yz \in L$, $xy \in E_{G'}$ or $xy \in S$,
(iv) the graph $H = (V_G, L)$ is such that $pw(H) \leq 2$.

Notice that the properties (i)–(iv) can be expressed in monadic second-order logic. In particular, (iv) can be expressed using the property that the class of graphs of pathwidth at most 2 is defined by the set of forbidden minors given by Kinnersley and Langston in [15]. Then we use Courcelle's theorem [5] to decide in linear time whether L exists or not.

To evaluate the running time, observe that to construct U, we consider each vertex $u \in V_G$ and check whether there are 5 distinct vertices in $N_G(u)$ that are at distance at least 3 from each other in $G - u$. This can be done in time $O(n^6)$ and implies that the total running time is also $O(n^6)$. $\square$

5 Conclusions

We proved that $\mathcal{H}$-Square Root is polynomial-time solvable when $\mathcal{H}$ is the class of outerplanar graphs or the class of graphs of pathwidth at most 2. The same result holds if $\mathcal{H}$ is any subclass of the class of outerplanar graphs that is closed under edge deletion and that can be expressed in monadic second-order logic (for instance, if $\mathcal{H}$ is the class of cactus graphs). We conclude by posing two questions:

– Is $\mathcal{H}$-Square Root polynomial-time solvable for every class $\mathcal{H}$ of graphs of bounded pathwidth?
– Is $\mathcal{H}$-Square Root polynomial-time solvable if $\mathcal{H}$ is the class of planar graphs?

References

1. Bodlaender, H.L.: A linear-time algorithm for finding tree-decompositions of small treewidth. SIAM Journal on Computing 25, 305–1317 (1996)
2. Bodlaender, H.L., Kloks, T.: Efficient and constructive algorithms for the pathwidth and treewidth of graphs. J. Algorithms 21(2), 358–402 (1996)
3. Cochefert, M., Couturier, J., Golovach, P.A., Kratsch, D., Paulusma, D.: Sparse square roots. In: Graph-Theoretic Concepts in Computer Science - 39th International Workshop, WG 2013. LNCS, vol. 8165, pp. 177–188. Springer (2013)
4. Cochefert, M., Couturier, J., Golovach, P.A., Kratsch, D., Paulusma, D.: Parameterized algorithms for finding square roots. Algorithmica 74, 602–629 (2016)
5. Courcelle, B.: The monadic second-order logic of graphs III: tree-decompositions, minor and complexity issues. Informatique Théorique et Applications 26, 257–286 (1992)
6. Courcelle, B., Engelfriet, J.: Graph Structure and Monadic Second-Order Logic - A Language-Theoretic Approach, Encyclopedia of mathematics and its applications, vol. 138. Cambridge University Press (2012)
7. Diestel, R.: Graph Theory, 4th Edition, Graduate Texts in Mathematics, vol. 173. Springer (2012)
8. Ducoffe G., Finding cut-vertices in the square roots of a graph, Proc. WG 2017, LNCS, to appear.
9. Farzad, B., Karimi, M.: Square-root finding problem in graphs, a complete dichotomy theorem. CoRR abs/1210.7684 (2012)
10. Farzad, B., Lau, L.C., Le, V.B., Tuy, N.N.: Complexity of finding graph roots with girth conditions. Algorithmica 62, 38–53 (2012)
11. Golovach, P.A., Heggernes, P., Kratsch, D., Lima, P.T., Paulusma, D.: Algorithms for outerplanar graph roots and graph roots of pathwidth at most 2. CoRR abs/1703.05102 (2017)
12. Golovach, P.A., Kratsch, D., Paulusma, D., Stewart, A.: Finding cactus roots in polynomial time. In: Proceedings of the 27th International Workshop on Combinatorial Algorithms (IWOCA 2016). LNCS, vol. 9843, pp. 253–265 (2016)
13. Golovach, P.A., Kratsch, D., Paulusma, D., Stewart, A.: A linear kernel for finding square roots of almost planar graphs. In: Proceedings of the 15th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2016). Leibniz International Proceedings in Informatics, vol. 53, pp. 4:1–4:14 (2016)
14. Golovach, P.A., Kratsch, D., Paulusma, D., Stewart, A.: Squares of low clique number. In: 14th Cologne Twente Workshop 2016 (CTW 2016). Electronic Notes in Discrete Mathematics, vol. 55, pp. 195–198 (2016)
15. Kinnersley, N.G., Langston, M.A.: Obstruction set isolation for the gate matrix layout problem. Discrete Applied Mathematics 54(2-3), 169–213 (1994)
16. Lau, L.C.: Bipartite roots of graphs. ACM Transactions on Algorithms 2, 178–208 (2006)
17. Lau, L.C., Corneil, D.G.: Recognizing powers of proper interval, split, and chordal graphs. SIAM Journal on Discrete Mathematics 18, 83–102 (2004)
18. Le, V.B., Oversberg, A., Schaudt, O.: Polynomial time recognition of squares of ptolemaic graphs and 3-sun-free split graphs. Theoretical Computer Science 602, 39–49 (2015)
19. Le, V.B., Oversberg, A., Schaudt, O.: A unified approach for recognizing squares of split graphs. Theoretical Computer Science 648, 26–33 (2016)

20. Le, V.B., Tuy, N.N.: The square of a block graph. Discrete Mathematics 310, 734–741 (2010)
21. Le, V.B., Tuy, N.N.: A good characterization of squares of strongly chordal split graphs. Information Processing Letters 111, 120–123 (2011)
22. Lin, Y., Skiena, S.: Algorithms for square roots of graphs. SIAM Journal on Discrete Mathematics 8, 99–118 (1995)
23. Milanic, M., Oversberg, A., Schaudt, O.: A characterization of line graphs that are squares of graphs. Discrete Applied Mathematics 173, 83–91 (2014)
24. Milanic, M., Schaudt, O.: Computing square roots of trivially perfect and threshold graphs. Discrete Applied Mathematics 161, 1538–1545 (2013)
25. Motwani, R., Sudan, M.: Computing roots of graphs is hard. Discrete Applied Mathematics 54, 81–88 (1994)
26. Mukhopadhyay, A.: The square root of a graph. Journal of Combinatorial Theory 2, 290–295 (1967)
27. Nestoridis, N.V., Thilikos, D.M.: Square roots of minor closed graph classes. Discrete Applied Mathematics 168, 34–39 (2014)
28. Ross, I.C., Harary, F.: The square of a tree. Bell System Technical Journal 39, 641–647 (1960)
29. Sysło, M.M.: Characterizations of outerplanar graphs. Discrete Math. 26(1), 47–53 (1979)