



HAL
open science

Finding cut-vertices in the square roots of a graph

Guillaume Ducoffe

► **To cite this version:**

Guillaume Ducoffe. Finding cut-vertices in the square roots of a graph. *Discrete Applied Mathematics*, 2019, 257, pp.158-174. hal-01948839

HAL Id: hal-01948839

<https://hal.science/hal-01948839>

Submitted on 8 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Finding cut-vertices in the square roots of a graph ¹

Guillaume Ducoffe^{a,b}

^a*National Institute for Research and Development in Informatics, Romania*

^b*The Research Institute of the University of Bucharest ICUB, Romania*

Abstract

The square of a given graph $H = (V, E)$ is obtained from H by adding an edge between every two vertices at distance two in H . Given a graph class \mathcal{H} , the \mathcal{H} -SQUARE ROOT PROBLEM asks for the recognition of the squares of graphs in \mathcal{H} . In this paper, we answer positively to an open question of [Golovach et al., IWOCA'16] by showing that the squares of *cactus-block graphs* can be recognized in polynomial time. Our proof is based on new relationships between the decomposition of a graph by cut-vertices and the decomposition of its square by clique cutsets. More precisely, we prove that the closed neighbourhoods of cut-vertices in H induce maximal subgraphs of $G = H^2$ with no clique-cutset. Furthermore, based on this relationship, we can compute from a given graph G the block-cut tree of a desired square root (if any). Although the latter tree is not uniquely defined, we show surprisingly that it can only differ marginally between two different roots. Our approach not only gives the first polynomial-time algorithm for the \mathcal{H} -SQUARE ROOT PROBLEM for several graph classes \mathcal{H} , but it also provides a unifying framework for the recognition of the squares of trees, block graphs and cactus graphs — among others.

Keywords: Square root of a graph; Clique-separator decomposition; Cut-vertices; Cactus-block graphs; Cycle-power graphs.

1. Introduction

This paper deals with the well-known concepts of *square* and *square root* in graph theory. Roughly, the square of a given graph is obtained by adding an edge between the pairs of vertices at distance two (technical definitions are postponed to Section 2). A square root of a given graph G has G as its square. The reason for this terminology is that when encoding a graph as an adjacency matrix A (with 1's on the diagonal), its square has for adjacency matrix A^2 — obtained from A using Boolean matrix multiplication. The squares of graphs appear, somewhat naturally, in the study of coloring problems: when it comes about modelling interferences at a bounded distance in a radio network [53]. Unsurprisingly, there is an important literature on the topic, with nice structural properties of square graphs being uncovered [2, 9, 21, 35, 38, 41]. In particular, an elegant characterization of the squares of graphs has been given in [43]. However, this does not lead to an efficient (polynomial-time) algorithm for recognizing square graphs. Our main focus in the paper is on the existence of such algorithms. They are, in fact, unlikely to exist since the problem has been proved to be NP-complete [42]. In light of this negative result, there has been

¹Results of this paper were partially presented at the WG'17 conference [18].

a growing literature trying to identify the cases where the recognition of the squares of graphs remains tractable [14, 27, 31, 30, 32, 37, 44]. We are interested in the variant where the desired square root (if any) must belong to some specified graph class.

1.1. Related work

There is a complete dichotomy result for the problem when it is parameterized by the *girth* of a square root. More precisely, the squares of graphs with girth at least six can be recognized in polynomial time, and it is NP-complete to decide whether a graph has a square root with girth at most five [1, 19, 20]. One first motivation for our work was to obtain similar dichotomy results based on the *separators* in a square root. We are thus more interested in graph classes with nice separability properties, such as chordal graphs. Recognizing the squares of chordal graphs is already NP-complete [31]. However, it can be done in polynomial time for many subclasses [31, 32, 33, 40, 45, 49].

The most relevant examples to explain our approach are the classes of trees [49], block graphs [33] and cacti [24]. The squares of all these graphs can be recognized in polynomial time. Perhaps surprisingly, whereas the case of trees is a well-known success story for which many algorithmic improvements have been proposed over the years [12, 33, 37, 49], the polynomial-time recognition of the squares of cactus graphs has been proved only very recently. A common point to these three above classes of graphs is that they can be decomposed into very simple subgraphs by using *cut-vertices* (respectively, in edges for trees, in complete graphs for block graphs and in cycles for cactus graphs). This fact is exploited in the polynomial-time recognition algorithms for the squares of these graphs. We observe that more generally, cut-vertices play a discrete, but important role, in the complexity of the recognition of squares, even for general graphs. As an example, most hardness results rely on a gadget called a “tail”, that is a particular case of cut-vertices in the square roots [20, 42]. Interestingly, this tail construction imposes for some vertex in the square to be a cut-vertex with the same closed neighbourhood in any square root (see Figure 1). It is thus natural to ask whether more general considerations on the cut-vertices can help to derive additional constraints on the closed neighbourhoods in these roots. Our results prove that it is the case.



Figure 1: Tail in a graph and its square.

As stated before, we are not the first to study the properties of cut-vertices in the square roots. In this respect, the work in [24] has been a major source of inspiration for this paper. However, most of the results so far obtained are specific to some graph classes and they hardly generalize to more general graphs [24, 33]. Evidence of this fact is that whereas both the squares of block graphs and the squares of cacti can be recognized in polynomial time, the techniques involved in these two cases do not apply to the slightly more general class of *cactus-block graphs* (graphs that can be decomposed by cut-vertices into cycles and complete graphs) [24]. In the end, the characterization

of the cut-vertices in these roots is only partial – even for cactus roots –, with most of the technical work for the recognition algorithm being rather focused on the notion of *tree decompositions* (e.g., clique-trees for chordal squares, or decomposition of the square into bounded-treewidth graphs). Roughly, tree decompositions [48] aim at decomposing graphs into pieces, called *bags*, organized in a tree-like manner. The decomposition of a square root of a graph by its cut-vertices leads to a specific type of tree decompositions for this graph that are called “*H*-tree decompositions” [23]. Note that it is not known whether a *H*-tree decomposition can be computed in polynomial time. In contrast, we use in this work another type of tree decompositions, called an *atom tree*, that generalizes the notion of clique-trees for every graph. It can be computed in polynomial time [5].

1.2. Our contributions

Our work is based on new relationships between the cut-vertices in a given graph and the *clique-cutsets* of its square (separators being a clique). These results are presented in Section 3. In particular, we obtain a complete characterization of the *atoms* of a graph (maximal subgraphs with no clique cutset) based on the *blocks* of its square roots (maximal subgraphs with no cut-vertices).

The most difficult part is to show how to “reverse” these relationships: from the square back to a square root. We prove in Section 4 that it can be done to some extent. More precisely, in Section 4.1 we show that the “essential” cut-vertices of the square roots, with at least two connected components not fully contained in their closed neighbourhoods, are in some sense unique (independent of the root) and that they can be computed in polynomial time, along with their closed neighbourhood in any square root. Indeed, structural properties of these vertices allow to reinterpret them as the cut-vertices of some incidence graphs that can be locally constructed from the intersection of the atoms in an atom tree (tree decomposition whose bags are exactly the atoms). Proving a similar characterization for non essential cut-vertices remains to be done. We give sufficient conditions and a complete characterization of the closed neighbourhoods of the non essential cut-vertices for a large class of graphs in Section 4.2.

Then, inspired from these above results, we introduce a novel framework in Section 5 for the recognition of squares. Assuming a square root exists, we can push further some ideas of Section 4 in order to compute, for every block in this root, a graph that is isomorphic to its square. This way, a square root can be computed for each square of a block separately. However, we need to impose additional constraints on these roots in order to be able to reconstruct from them a square root for the original graph. We thus reduce the recognition of the squares to a stronger variant of the problem for the squares of biconnected graphs. Let us point out that this approach can be particularly beneficial when the blocks of a root are assumed to be part of a well-structured graph class (e.g., regular bounded treewidth graphs with some additional properties). We detail sufficient conditions for the framework to be applied in Section 5.3.

In Section 6, we finally answer positively to an open question of [24] by proving that the squares of cactus-block graphs can be recognized in polynomial time. Our result is actually much more general, as it gives a unifying algorithm for many graph classes already known to be tractable (e.g., trees, block graphs and cacti) and it provides the first polynomial time recognition algorithm for the squares of related graph classes – such as Gallai trees [22]. In its full generality, the result applies to “*j*-cactus-block graphs”: a generalization of cactus-block graphs where each block is either a complete graph or the k^{th} -power of a cycle, for some $1 \leq k \leq j$. As expected this last result is obtained by using our framework. This application is not straightforward. Indeed, we need to show the existence of a *j*-cactus-block root with some “good” properties in order for the framework to

be applied. We also need to show that a stronger variant of the recognition of squares (discussed in Section 5) can be solved in polynomial time for j -cactus-block graphs when j is a fixed constant. We do so by introducing classical techniques from the study of circular-arc graphs [52].

Although we keep the focus on square roots, we think that our approach could be generalized in order to compute the cut-vertices in the p^{th} -roots of a graph (e.g., see [12] for related work on p^{th} -tree roots). This is left for future work. Definitions and preliminary results are given in Section 2. We conclude this paper in Section 7 with some open questions.

2. Preliminaries

We use standard graph terminology from [10]. All graphs in this study are finite, unweighted and simple (hence with neither loops nor multiple edges), unless stated otherwise. Given a graph $G = (V, E)$ and a set $S \subseteq V$, we will denote by $G[S]$ the subgraph of G that is induced by S . The open neighbourhood of S , denoted by $N_G(S)$, is the set of all vertices in $G[V \setminus S]$ that are adjacent to at least one vertex in S . Similarly, the closed neighbourhood of S is denoted by $N_G[S] = N_G(S) \cup S$. For every $u, v \in V$, vertex v is *dominated*² by u if $N_G[v] \subseteq N_G[u]$. In particular, if $N_G[u] = N_G[v]$ then we say u and v are *true twins*. If even more strongly, we have $N_G[w] \subseteq N_G[u]$ for every $w \in N_G[v]$, then u is a *maximum neighbour* of v . A *pending vertex* of G is a vertex of degree one. The unique neighbour of a pending vertex v is always a maximum neighbour of v .

2.1. Squares and powers of graphs

For every connected graph G and for every $u, v \in V$, the distance between u and v in G , denoted by $\text{dist}_G(u, v)$, is equal to the minimum length (number of edges) of a uv -path in G . The j^{th} -power of G is the graph $G^j = (V, E_j)$ with same vertex-set as G and an edge between every two distinct vertices at distance at most j in G . In particular, the *square* of a graph $G = (V, E)$ is the graph $G^2 = (V, E_2)$ with same vertex-set V as G and an edge between every two distinct vertices $u, v \in V$ such that $N_G[u] \cap N_G[v] \neq \emptyset$. Conversely, if there exists a graph H such that G is isomorphic to H^2 then H is called a *square root* of G . On the one hand it is easy to see that not all graphs have a square root. For example, if G is a tree with at least three vertices then it does not have any square root. On the other hand, note that a graph can have more than one square root. As an example, the complete graph K_n with n vertices is the square of any diameter two n -vertex graph.

We focus on the following recognition problem:

Problem 1 (\mathcal{H} -SQUARE ROOT).

Input: A graph $G = (V, E)$.

Question: Is G the square of a graph in the class \mathcal{H} ?

Our proofs will make use of the notions of subgraphs, induced subgraphs and *isometric subgraphs*, the latter denoting a subgraph H of a connected graph G such that $\text{dist}_H(x, y) = \text{dist}_G(x, y)$ for every $x, y \in V(H)$. Furthermore, let H be a square root of a given graph $G = (V, E)$. Given a walk $\mathcal{W} = (x_0, x_1, \dots, x_\ell)$ in G , an *H-extension* of \mathcal{W} is any walk \mathcal{W}' of H that is obtained

² Although it is standard, this terminology is slightly confusing as it does not coincide with the classical notion of domination in graphs (i.e., one vertex is in the closed neighborhood of another).

from \mathcal{W} by adding, for every i such that x_i and x_{i+1} are nonadjacent in H , a common neighbour $y_i \in N_H(x_i) \cap N_H(x_{i+1})$ between x_i and x_{i+1} .

2.2. Graph decompositions

A set $S \subseteq V$ is a *separator* in a graph $G = (V, E)$ if its removal increases the number of connected components³. A *full component* in $G[V \setminus S]$ is any connected component C in $G[V \setminus S]$ satisfying that $N_G(C) = S$ (note that a full component might fail to exist). The set S is called a *minimal separator* in G if it is a separator and there are at least two full components in $G[V \setminus S]$. Minimal separators are closely related to the notion of Robertson and Seymour's *tree decompositions* (e.g., see [11, 25, 28, 46]). Formally, a *tree-decomposition* (T, \mathcal{X}) of G is a pair consisting of a tree T and of a family $\mathcal{X} = (X_t)_{t \in V(T)}$ of subsets of V indexed by the nodes of T and satisfying:

- $\bigcup_{t \in V(T)} X_t = V$;
- for any edge $e = \{u, v\} \in E$, there exists $t \in V(T)$ such that $u, v \in X_t$;
- for any $v \in V$, $\{t \in V(T) \mid v \in X_t\}$ induces a subtree, denoted by T_v , of T .

The sets X_t are called the *bags* of the decomposition.

In what follows, we will consider two main types of minimal separators.

Cut-vertices. If $S = \{v\}$ is a separator then it is a minimal one and we call it a *cut-vertex* of G . Following the terminology of [24], we name v an *essential* cut-vertex if there are at least two components C_1, C_2 of $G \setminus v$ such that $C_1 \not\subseteq N_G(v)$ and similarly $C_2 \not\subseteq N_G(v)$; otherwise, v is called a *non essential* cut-vertex⁴. A graph $G = (V, E)$ is *biconnected* if it is connected and it does not have a cut-vertex. Examples of biconnected graphs are cycles and complete graphs. Furthermore, the *blocks* of G are the maximal biconnected subgraphs of G . For every connected graph G there is a tree whose nodes are the blocks and the cut-vertices of G , sometimes called the *block-cut tree*, that is obtained by adding an edge between every block B and every cut-vertex v such that $v \in V(B)$ (see Figure 2 for an example). The block-cut tree of a given connected graph G can be computed in linear time [29].

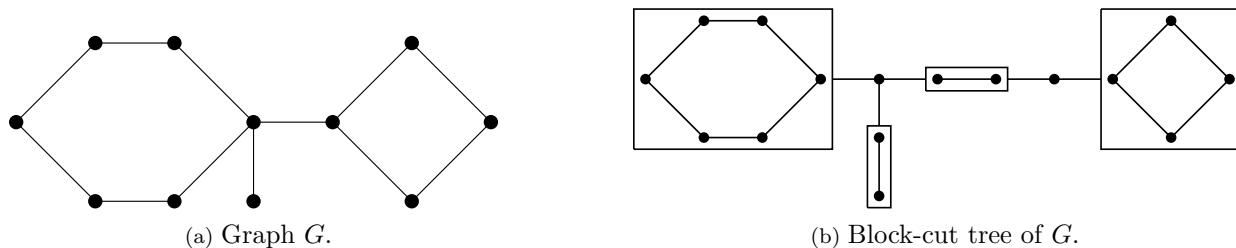


Figure 2: An example of block-cut-tree.

³ Unlike some authors we consider a a connected component as a vertex-subset and not as a subgraph.

⁴ The authors in [24] have rather focused on the stronger notion of *important* cut-vertices, that requires the existence of an additional third component C_3 of $G \setminus v$ such that $C_3 \not\subseteq N_G(v)$. We do not use this notion in our paper.

It has been observed that every graph with a square root is biconnected [21]. We often use this fact in what follows. Furthermore, the following was proved in [23].

Lemma 1 ([23]). *Let H be a square root of a graph G . Let \mathcal{C} and \mathcal{B} be the sets of cut-vertices and blocks of H , respectively, and let T_H be the block-cut tree of H , with vertex set $V_T = \mathcal{C} \cup \mathcal{B}$. For every $u \in \mathcal{C}$, let $X_u = N_H[u]$ and for every $B \in \mathcal{B}$, let $X_B = V(B)$. Then, $(T_H, (X_t)_{t \in V_T})$ is a tree decomposition of G that is called the H -tree decomposition of G .*

Clique cutsets. More generally, if S is a minimal separator inducing a complete subgraph of $G = (V, E)$ then we call it a *clique cutset* of G . A connected graph $G = (V, E)$ is *prime* if it does not have a clique cutset. Cycles and complete graphs are again examples of prime graphs, and it can be observed more generally that every prime graph is biconnected. The *atoms* of G are the maximal prime subgraphs of G . They can be computed in polynomial time [34, 51]. A *clique-atom* is an atom inducing a complete subgraph. Furthermore, a *simplicial vertex* is a vertex $v \in V$ such that $N_G[v]$ is a clique. The following was proved in [15].

Lemma 2 ([15]). *Let $G = (V, E)$ be a graph and let $v \in V$. Then, v is simplicial if and only if $N_G[v]$ is a clique-atom, and it is the unique atom containing v .*

If the atoms of G are given, then the clique-atoms and the simplicial vertices of G can be computed in linear time [15].

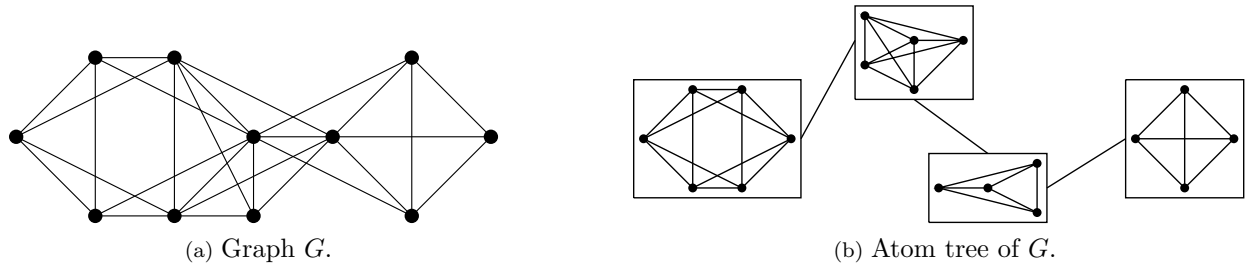


Figure 3: An example of atom tree.

Finally, it has been proved in [5] that there exists a tree decomposition of G of which the bags are exactly the atoms of G . Such a tree decomposition is sometimes called an *atom tree*. We refer to Figure 3 for an illustration. An atom tree can be computed in $\mathcal{O}(nm)$ -time, and it is not necessarily unique [5]. In what follows, we often use in our analysis the following properties of an atom tree:

Lemma 3 ([6]). *Let $G = (V, E)$ be a graph and let A, A' be atoms of G . Then, $A \setminus A'$ is a connected subset and $A \cap A' \subseteq N_G(A \setminus A')$.*

Lemma 4 ([6]). *Let $G = (V, E)$ be a graph. For every atom tree (T_G, \mathcal{A}) of G , we have:*

$$\sum_{\{A, A'\} \in E(T_G)} |A \cap A'| = \mathcal{O}(n + m).$$

3. Basic properties of the atoms in a square

We start presenting relationships between the block-cut tree of a given graph and the decomposition of its square by clique cutsets (Theorem 6). These relationships are compared after the

proof to some existing results in the literature for the \mathcal{H} -SQUARE ROOT problem. More precisely, our approach in this paper is based on the following relationship between the clique cutsets in a graph G and the cut-vertices in its square-roots (if any).

Proposition 5. *Let $H = (V, E)$ be a graph. The closed neighbourhood of any cut-vertex in H is a clique-atom of $G = H^2$.*

Proof. Let $v \in V$ be a cut-vertex of H and let $A_v = N_H[v]$. It is clear that A_v is a clique of G and so, this set induces a prime subgraph of G . In particular, A_v must be contained in an atom A of G . Suppose for the sake of contradiction that $A \neq A_v$. Let $u \in A \setminus A_v$. This vertex u is contained in some connected component C_u of $H \setminus v$. Furthermore since v is a cut-vertex of H , there exists $w \in N_H(v) \setminus C_u$. We claim that $S = (C_u \cap N_H(v)) \cup \{v\}$ is an uw -clique cutset of G . Indeed, let us consider any uw -path \mathcal{P} in G . We name $\mathcal{Q} = (x_0 = u, x_1, \dots, x_\ell = w)$ an arbitrary H -extension of \mathcal{P} . Since \mathcal{Q} is an uw -walk in H , and u and w are in different connected components of $H \setminus v$, there exists an i such that $x_i \in C_u$, $x_{i+1} = v$. In particular, $x_i \in C_u \cap N_H(v) = S \setminus \{v\}$. Furthermore, by construction, for every two consecutive vertices x_i, x_{i+1} in the H -extension \mathcal{Q} , at least one of x_i or x_{i+1} belongs to \mathcal{P} . As a result, every uw -path in G intersects S , that proves the claim and so, that contradicts the fact that A is an atom of G . Therefore, $A = A_v$. Since A_v is a clique it is indeed a clique-atom of G . \square

The above Proposition 5 unifies and generalizes some previous results that have been found only for specific graph classes [24, 33]. For example, it has been proved in [33] that for every block-graph H , the closed neighbourhoods of its cut-vertices are maximal cliques of its square. Our result shows that it holds for *any* square root H (not only block-graphs). Indeed, a clique-atom is always a maximal clique. Furthermore, our purpose with Theorem 6 is to give a partial characterization of the remaining atoms of the square. Ideally, we would have liked them to correspond to the blocks of its square roots. It turns out that this is not always the case. A counter-example is given in Figure 4. However, as we prove next there are strong ties between the two.

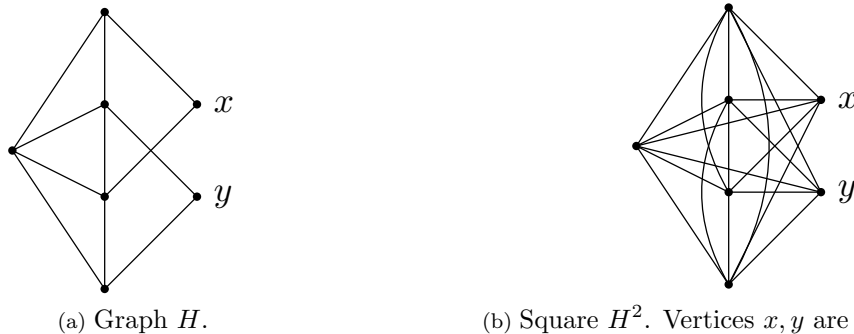


Figure 4: A biconnected graph with a non prime square.

Theorem 6. *Let H be the square root of a given graph $G = (V, E)$. Then, the set of all the atoms of G is the disjoint union of:*

- the cliques $A_v = N_H[v]$, for every cut-vertex v of H ;
- and for every block B of H , the atoms A' of B^2 that are not contained into the closed neighbourhood of a cut-vertex in H .

Proof. Let A be an atom of G . There are two cases to be distinguished.

1. Suppose that A is not contained in a block of H . Let B_1, B_2 be two different blocks of H intersecting A and let $u \in V(B_1 \setminus B_2)$, $w \in V(B_2 \setminus B_1)$ such that $u, w \in A$. By the hypothesis, there exists a cut-vertex v of H that intersects every uw -path of H . We claim that $A = A_v$. Indeed, suppose for the sake of contradiction that $A \neq A_v$. W.l.o.g., $u \notin A_v$. Let us name by C_u the connected component of $H \setminus v$ that contains u . Then, it is easy to prove (as we did for Proposition 5) that every uw -path in G must intersect the clique $S = (N_H(v) \cap C_u) \cup \{v\}$. The latter contradicts our assumption that the atom A has no clique cutset. Therefore, we obtain that $A \subseteq A_v$. Since A_v is an atom by Proposition 5, it follows as claimed that $A = A_v$.
2. Else, $A \subseteq V(B)$, for some block B of H . Observe that B is an isometric subgraph of H , hence B^2 and $G[V(B)]$ are isomorphic. As a result, A is an atom of G implies that A is also an atom of B^2 . Conversely, if A' is an atom of B^2 then A' induces a prime subgraph of G . In particular, it is an atom of G if and only if it is inclusion wise maximal w.r.t. this property. Finally, we note that the only atoms that can possibly contain A' are the sets A_v , for any cut-vertex v of H that is contained in $V(B)$. Altogether combined, we have in this situation that A is an atom of B^2 such that, for every cut-vertex v of H that is contained in $V(B)$, $A \not\subseteq A_v$.

□

From Theorem 6, we can deduce that the atom trees of a given square graph G are related to its *H-tree decompositions*, as introduced in Lemma 1. Indeed, for any square root H of G , the atom trees of G are refinements of its *H-tree decomposition*, in the sense that every atom is included in a bag of the *H-tree decomposition* of G . Furthermore, the atoms of G coincide with the bags of its *H-tree decomposition* if and only if: every block of H has a prime square, and there is no block of H included in the closed neighbourhood of a cut-vertex. In this situation, the *H-tree decomposition* of G is an atom tree. However, it may be the case that G has other atom trees. We refer to [50] for counting the number of atom trees for a given graph.

4. Computation of the cut-vertices from the square

Given a square graph $G = (V, E)$, we aim at computing all the cut-vertices in some square root H of G . More precisely, given two square roots H_1 and H_2 of G , we say that H_1 is “finer” than H_2 if the blocks of H_1 are contained in the blocks of H_2 . The latter defines a partial ordering over the square roots of G , of which we call *maxblock square roots* its minimal elements. For convenience, we also introduce the notion of *finest square root* in a class \mathcal{H} , defined as a square root $H \in \mathcal{H}$ of G such that no other square root of G in \mathcal{H} can be finer than H . This notion is related to, but different than, the notion of minimal square root studied in [24]⁵. As an example, maxblock square roots are exactly the finest square roots in the class of all graphs.

The following section is based on Proposition 5, that gives a necessary condition for a vertex to be a cut-vertex in any maxblock square root H_{\max} of G . Indeed, it follows from this Proposition 5 that there is a mapping from the cut-vertices of H_{\max} to the clique-atoms of its square $G = H_{\max}^2$.

⁵ Let \mathcal{H} be a graph class closed under edge deletion. If G has a square root in \mathcal{H} then there exists a finest square root $H \in \mathcal{H}$ such that H is a minimal square root of G .

We claim that this mapping is injective. Indeed, it follows from the existence of the block-cut tree that every two distinct cut-vertices v, v' of H_{\max} can be contained in at most one common block. Since every cut-vertex is contained in at least two blocks, we get $N_{H_{\max}}[v] \neq N_{H_{\max}}[v']$, thereby proving the claim. However, the mapping is not surjective in general.

In what follows, we present sufficient conditions for a clique-atom of G to be the closed neighbourhood of a cut-vertex in any maxblock square root of G . In particular, we obtain a complete characterization for the essential cut-vertices.

4.1. Recognition of the essential cut-vertices

We recall that a cut-vertex v of H_{\max} is called essential if there are two vertices in different connected components of $H_{\max} \setminus v$ that are both at distance two from v in H_{\max} . The remaining of the section is devoted to prove the following result.

Theorem 7. *Let $G = (V, E)$ be a square graph. Every maxblock square root of G has the same set \mathcal{C} of essential cut-vertices. Furthermore, every vertex $v \in \mathcal{C}$ has the same neighbourhood A_v in any maxblock square root of G . All the vertices $v \in \mathcal{C}$ and their neighbourhood A_v can be computed in $\mathcal{O}(n + m)$ -time if an atom tree of G is given.*

Algorithm 1 Computation of the essential cut-vertices.

Require: A graph $G = (V, E)$; an atom tree (T_G, \mathcal{A}) of G .

Ensure: Returns (if G is a square) the set \mathcal{C} of essential cut-vertices, and for every $v \in \mathcal{C}$ its neighbourhood A_v , in any maxblock square root of G .

- 1: $\mathcal{C} \leftarrow \emptyset$.
 - 2: **for all** clique-atom $A \in \mathcal{A}$ **do**
 - 3: Compute the incidence graph $I_A = \text{Inc}(\Omega(A), A)$, with $\Omega(A)$ being the multiset of neighbourhoods of the connected components of $G \setminus A$.
 - 4: **if** $\bigcap_{S \in \Omega(A)} S = \{v\}$ **and** v is a cut-vertex of I_A **then**
 - 5: $\mathcal{C} \leftarrow \mathcal{C} \cup \{v\}$; $A_v \leftarrow A$.
-

The proof of Theorem 7 mainly follows from the correctness proof and the complexity analysis of Algorithm 1. Its basic idea is that the essential cut-vertices in any maxblock square root of G are exactly the cut-vertices in some “incidence graphs”, that are locally constructed from the neighbourhoods of each clique-atom in the atom tree. Formally, for every clique-atom A of G , let $\Omega(A)$ contain $N_G(C)$ for every connected component C of $G \setminus A$ (note that $\Omega(A)$ is a multiset, with its cardinality being equal to the number of connected components in $G \setminus A$). The incidence graph $I_A = \text{Inc}(\Omega(A), A)$ is the bipartite graph with respective sides $\Omega(A)$ and A and an edge between every $S \in \Omega(A)$ and every $u \in S$.

We first need to observe that for every $v \in A$, v is a cut-vertex of I_A if and only if there is a bipartition P, Q of the connected components of $G \setminus A$ such that $N_G(P) \cap N_G(Q) = \{v\}$. Then, we subdivide the correctness proof of Algorithm 1 in two lemmas.

Lemma 8. *Let H be a square root of a given graph $G = (V, E)$, let $v \in V$ be an essential cut-vertex of H and let $A_v = N_H[v]$. Then, v has a neighbour in G in every connected component of $G \setminus A_v$. Furthermore, there is a bipartition P, Q of the connected components of $G \setminus A_v$ such that $N_G(P) \cap N_G(Q) = \{v\}$.*

Proof. First, observe that for every connected component D of $G \setminus A_v$, we have that $N_H(D) \cap A_v \neq \emptyset$. Since $A_v = N_H[v]$, it follows that $v \in N_G(D)$. In particular, v has a neighbour in G in every connected component of $G \setminus A_v$. Second, let C_1, C_2, \dots, C_k be all the connected components of $H \setminus v$ such that $C_i \not\subseteq A_v$. Note that $k \geq 2$ since by the hypothesis v is an essential cut-vertex of H . Furthermore, since for every $i \neq j$ and for every $u_i \in C_i \setminus A_v$, $u_j \in C_j \setminus A_v$, we have $\text{dist}_H(u_i, u_j) = \text{dist}_H(u_i, v) + \text{dist}_H(u_j, v) \geq 4$, there can be no edge between $C_i \setminus A_v$ and $C_j \setminus A_v$ in G . It implies that for every component D of $G \setminus A_v$, there is an $1 \leq i \leq k$ such that $D \subseteq C_i \setminus A_v$. So, we can partition $G \setminus A_v$ into the sets $C_i \setminus A_v$, $1 \leq i \leq k$, and this also defines a partition of the components of $G \setminus A_v$. For every $1 \leq i \leq k$, we have $\{v\} \subseteq N_G(C_i \setminus A_v) \subseteq (N_H(v) \cap C_i) \cup \{v\}$. In particular, for every $i \neq j$, we obtain $N_G(C_i \setminus A_v) \cap N_G(C_j \setminus A_v) = \{v\}$. Hence, let us bipartition the sets $C_i \setminus A_v$ into two nonempty supersets P and Q ; by construction we have $N_G(P) \cap N_G(Q) = \{v\}$. \square

It turns out that conversely, Lemma 8 also provides a sufficient condition for a vertex v to be an essential cut-vertex in some square root of G (and in particular, in any maxblock square root). We formalize this next.

Lemma 9. *Let H_{\max} be a maxblock square root of a given graph $G = (V, E)$, and let $v \in V$. Assume that there is a clique-atom A_v of G and a bipartition P, Q of the connected components of $G \setminus A_v$ such that $N_G(P) \cap N_G(Q) = \{v\}$. Then, for every square root H of G , we have $N_H(P) \cup N_H(Q) \subseteq N_H(v) \subseteq A_v$. In particular, v is an essential cut-vertex of H_{\max} and $N_{H_{\max}}[v] = A_v$.*

Proof. Let H be any square root of G . By the hypothesis P, Q is a bipartition of the connected components of $G \setminus A_v$, hence there can be no edge in G between P and Q , and so, $N_H(P) \cap N_H(Q) = \emptyset$. Furthermore, if it were the case that $u \in N_H(P)$ and $w \in N_H(Q)$ are adjacent in H , then it would imply that $u, w \in N_G(P) \cap N_G(Q)$, that would contradict the hypothesis that $N_G(P) \cap N_G(Q) = \{v\}$. As a result, since A_v is a clique of G we have that for every $u \in N_H(P)$ and $w \in N_H(Q)$, there exists a common neighbour $v' \in N_H(u) \cap N_H(w)$, and necessarily $v' \notin N_H(P) \cup N_H(Q)$ (otherwise it would imply the existence of an edge in H between $N_H(P)$ and $N_H(Q)$). In particular, $N_H(v') \subseteq A_v$. Since in addition $v' \in N_G(P) \cap N_G(Q)$ then it must be the case that $v' = v$. Altogether combined, it implies that $N_H(P) \cup N_H(Q) \subseteq N_H(v) \subseteq A_v$.

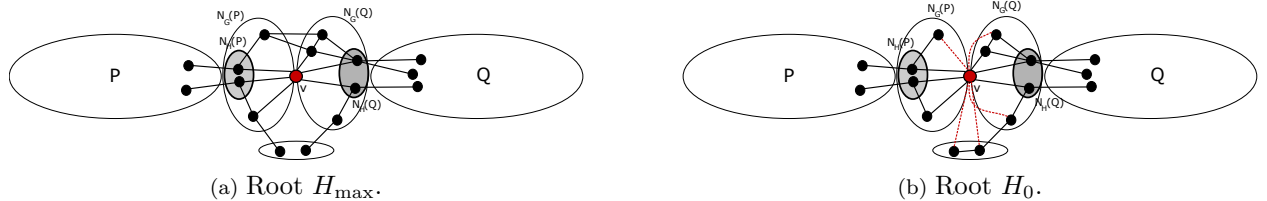


Figure 5: The transformation of Lemma 9. Some vertices and edges are omitted.

Then, suppose for the sake of contradiction that v is not a cut-vertex of H_{\max} . We claim that A_v must be contained in a block of H_{\max} . Indeed, observe that v is the unique vertex such that $N_{H_{\max}}(P) \cup N_{H_{\max}}(Q) \subseteq N_{H_{\max}}(v)$ (i.e., because $N_{H_{\max}}(P) \cup N_{H_{\max}}(Q) \subseteq N_{H_{\max}}(v') \implies v' \in N_G(P) \cap N_G(Q)$). Hence, there can be no cut-vertex of H_{\max} whose closed neighbourhood is A_v . In this situation, by Theorem 6 A_v is an atom in the square of some block B of H_{\max} . It implies as claimed that A_v must be contained in B . Let H_0 be obtained from H_{\max} as follows (see also Fig. 5). For every vertex $u \in N_G(P) \setminus \{v\}$ we first remove all the edges in H_{\max} between u and every vertex of $A_v \setminus N_G(P)$. Note that this operation does not change the neighbourhood (the

neighbourhood at distance two, resp.) of P and Q . Then, we make every vertex of A_v adjacent to v . By doing so, we claim that we make of vertex v a cut-vertex in H_0 . In order to prove the claim, it suffices to prove that there is no path between P and Q in $H_0 \setminus v$. Suppose for the sake of contradiction there exists such a path. As P, Q is a bipartition of the connected components of $G \setminus A_v$, there should exist a path between P and Q where all the internal vertices are in A_v . In particular, since $N_G(P) \setminus \{v\}$ and $N_G(Q) \setminus \{v\}$ are disjoint, there should be an edge between $N_G(P) \setminus \{v\}$ and $A_v \setminus N_G(P)$, thereby contradicting our construction of H_0 , and so, proving the claim. We so increase the number of blocks. Furthermore, A_v is still a clique of H_0^2 , and since $N_{H_{\max}}(P) \cup N_{H_{\max}}(Q) \subseteq N_{H_{\max}}(v)$, the adjacency relations between the vertices of $P \cup Q$ and the vertices of A_v are the same in H_0^2 as in G . Hence, by construction H_0 is a square root of G which is finer than H_{\max} , that contradicts the fact that H_{\max} is a maxblock square root. Therefore, v is a cut-vertex of H_{\max} . By using the same arguments as above, it can also be proved that the vertices in P are in different connected components of $H_{\max} \setminus v$ than the vertices in Q . Hence, v is an essential cut-vertex of H_{\max} . Finally, since $N_{H_{\max}}[v] \subseteq A_v$ and $N_{H_{\max}}[v]$ is a clique-atom of G by Proposition 5, we have $N_{H_{\max}}[v] = A_v$. \square

Correctness of Algorithm 1 follows from Lemmas 8 and 9. In order to obtain a linear-time implementation, we replace the incidence graph I_A with a “reduced version” I_A^* , where we only consider the adhesion sets in an atom tree of G (intersection of A with the adjacent atoms in the atom tree). Indeed, doing so we simply discard the neighbourhoods of some components that are strictly contained in the neighbourhood of another component. Using the fact that G is biconnected, it can be shown that this does not affect the outcome. This allows us to achieve a time complexity that is linear in the size of the atom tree, and so, linear in the size of the input graph G .

Proof of Theorem 7. The unicity of \mathcal{C} and of the sets A_v , $v \in \mathcal{C}$ follows from Lemmas 8 and 9. Indeed, on the one hand let $v \in V$ be an essential cut-vertex in some square root H of G and let $A_v = N_H[v]$. Let $I_{A_v} = Inc(\Omega(A_v), A_v)$ be the incidence graph as defined above. By Lemma 8 we have $v \in \bigcap_{S \in \Omega(A)} S$. Furthermore, since v is a cut-vertex of I_A , it is the unique vertex in this

common intersection $\bigcap_{S \in \Omega(A)} S$. As a result, vertex v passes the test of Algorithm 1 for $A = A_v$.

Conversely, if for some clique-atom A the (unique) vertex v tested passes the test of Algorithm 1, then by Lemma 9 v is an essential cut-vertex with closed neighbourhood being equal to A in any maxblock square root of G .

Let us finally prove that given an atom tree (T_G, \mathcal{A}) of G , the set \mathcal{C} and the closed neighbourhood A_v , $v \in \mathcal{C}$ can be computed in linear time. In order to prove it, it suffices to prove that Algorithm 1 can be implemented to run in linear time. We first remind that the set of clique-atoms of G can be computed from (T_G, \mathcal{A}) in linear time [15].

Furthermore, let $A \in \mathcal{A}$ be a fixed clique-atom, and let $\Omega^*(A) = \{A' \cap A \mid \{A', A\} \in E(T_G)\}$. By the properties of atom trees (Lemma 3), $\Omega^*(A)$ is a subset of $\Omega(A)$. In addition (by the properties of tree decompositions) we have that every set $S \in \Omega(A)$ is included in some set $S^* \in \Omega^*(A)$. So, instead of computing the incidence graph I_A , let us compute the “reduced incidence graph” $I_A^* = Inc(\Omega^*(A), A)$. Let us replace I_A by I_A^* for the test of Algorithm 1. By doing so, this test can be performed in $\mathcal{O}(|E(I_A^*)|) = \mathcal{O}(\sum_{S^* \in \Omega^*(A)} |S^*|)$.

Let us prove that this modification of the test does not change the output of the algorithm. On the one hand, if $v \in A$ passes the modified test then by Lemma 9, $v \in \mathcal{C}$ and $A_v = A$. Conversely, let $v \in \mathcal{C}$ be arbitrary and let $A = A_v$. Note that since by Lemma 8 we have $v \in \bigcap_{S \in \Omega(A)} S$, we

also have $v \in \bigcap_{S^* \in \Omega^*(A)} S^*$. In addition, since G is assumed to be a square, it is biconnected and so,

$|S| \geq 2$ for every $S \in \Omega(A)$; in particular, every two $S, S^* \in \Omega(A)$ such that $S \subseteq S^*$ are in the same block of I_A (since they have two common neighbours in I_A), that implies that v keeps the property to be a cut-vertex of I_A^* . The latter implies as before that $\bigcap_{S^* \in \Omega^*(A)} S^* = \{v\}$. It thus follows that

every vertex $v \in \mathcal{C}$ passes the modified test.

Overall, this above implementation of Algorithm 1 runs in time:

$$\mathcal{O}\left(\sum_{A \in \mathcal{A}} \sum_{S^* \in \Omega^*(A)} |S^*|\right) = \mathcal{O}\left(\sum_{\{A, A'\} \in E(T_G)} |A \cap A'|\right).$$

Since (T, \mathcal{A}) is an atom tree, it is $\mathcal{O}(n + m)$ by Lemma 4. □

4.2. Sufficient conditions for non essential cut-vertices

We left open whether a good characterization of non essential cut-vertices can be found. The remaining of this section is devoted to partial results in this direction. In general, not all the maxblock square roots of a graph have the same set of non essential cut-vertices. Indeed, a non essential cut-vertex v can have a true twin u in the square graph, in which case there exists another square root where the roles of u and v have been exchanged (*e.g.*, see Figure 6).

Our main result in this section is a complete characterization of the closed neighbourhoods of such vertices in any *finest* square root with some prescribed properties being satisfied by its blocks (Theorem 11). Admittedly, these additional conditions are a bit technical. However, we show in the next sections that they are satisfied by many interesting graph classes.

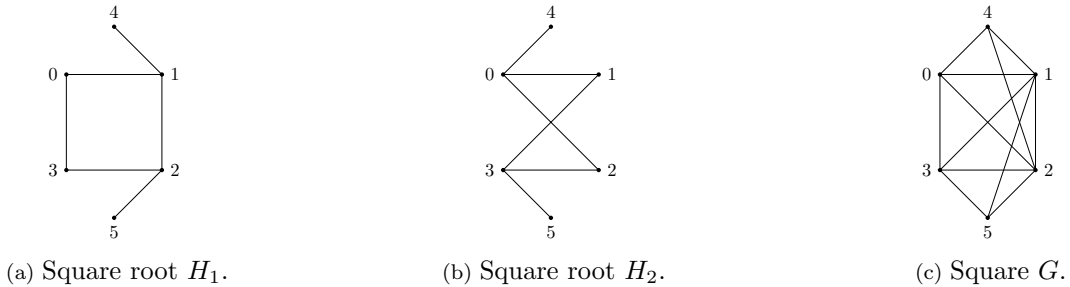


Figure 6: Two non isomorphic square roots where the cut-vertices are different.

Non essential cut-vertices in a maxblock square root are strongly related to simplicial vertices in the square. In general, if a clique-atom of G contains a simplicial vertex then it may not necessarily represent the closed neighbourhood of a cut-vertex in H_{\max} . However, we can prove it is always the case if the vertex is *simple*, *i.e.*, it is simplicial and the closed neighbourhoods of its neighbours can be linearly ordered by inclusion.

Lemma 10. *Let H_{\max} be a maxblock square root of a graph $G = (V, E)$. If there exists a simple vertex u in G then it has a neighbour $v \in N_G(u)$ that is a non essential cut-vertex of H_{\max} . Furthermore, $N_{H_{\max}}[v] = N_G[u]$.*

Proof. Let $A = N_G[u]$. Since u is simple, and so, simplicial, by Lemma 2 A is the unique atom of G containing u . Furthermore, we claim that A is the closed neighbourhood of a cut-vertex in H_{\max} . By contradiction, suppose that it is not the case. By Theorem 6, A is an atom in the square of some block of H_{\max} . Hence, A is contained in a block of H_{\max} . We will transform H_{\max} so that it is no longer the case, that will arise a contradiction. In order to do so, let $S = S_1 \cup S_2$ with $S_1 = N_{H_{\max}}(V \setminus A)$, $S_2 = N_{H_{\max}}(S_1) \cap A$ and let $K = A \setminus S$. Observe that $K \neq \emptyset$ (since $u \in K$). Thus, since $N_{H_{\max}}(K) \subseteq S_2$ and A is a clique-atom of G , we have that $N_{H_{\max}}(w) \cap S_2 \neq \emptyset$ for every $w \in S_1$. In particular, $\bigcup_{v' \in S_2} N_G(v') \setminus A = \bigcup_{w \in S_1} N_{H_{\max}}(w) \setminus A = N_{H_{\max}}(A)$.

Let v be among all the vertices in S_2 one of maximum degree in G . Note that since u is assumed to be simple by the hypothesis, $N_G[v'] \subseteq N_G[v]$ for every $v' \in S_2$. So, let H_0 be obtained from H_{\max} by removing all the edges incident to a vertex in K , then making vertex v universal in A . By construction, all the vertices in K are now pending vertices adjacent to v . Furthermore, H_0 has strictly more cut-vertices and strictly more blocks than H_{\max} . We claim that H_0 is a square root of G , that will arise a contradiction. Indeed, on the one hand A is a clique of H_0^2 . In addition, for every $v' \in S \setminus \{v\}$ we have $N_{H_0^2}(v') \setminus A = N_G(v') \setminus A$ because $N_{H_{\max}}[v'] \cap S_1 = N_{H_0}[v'] \cap S_1$. On the

other hand, $N_{H_0^2}(v) \setminus A = N_{H_{\max}}(S_1) \setminus A = N_{H_{\max}}(A) = \left(\bigcup_{v' \in S_2} N_G(v') \right) \setminus A = N_G(v) \setminus A$. As a result, H_0 is a square root of G which is finer than H_{\max} , thereby contradicting the fact that H_{\max} is a maxblock square root of G . This implies as claimed that A is the closed neighbourhood of a cut-vertex v of H_{\max} .

In order to complete the proof of the lemma, it suffices to prove that v cannot be an essential cut-vertex. Indeed, let us totally order the vertices $w_1, w_2, \dots, w_k \in S$ such that $i < j \implies N_G[w_i] \subseteq N_G[w_j]$ (it can be done since u is simple). In this situation, if for some component C of $G \setminus A$, we have $w_i \in N_G(C)$, then we also have $w_i, w_{i+1}, \dots, w_k \in N_G(C)$. In particular, since G is biconnected, it implies that w_{k-1}, w_k have a neighbour in every component of $G \setminus A$. As a result, there can be no partition P, Q of the connected components of $G \setminus A$ such that $N_G(P) \cap N_G(Q)$ is reduced to a singleton. It thus follows from Lemma 8 that A cannot be the closed neighbourhood of an essential cut-vertex. \square

Before concluding this section, we now state its main result.

Theorem 11. *Let $G = (V, E)$ be a connected graph that is not a complete graph. Furthermore, let H_{\max} be a finest square root of G with the property that, for every block B of H_{\max} , we have:*

- B has no dominated vertex, unless it is a clique;
- and B^2 is prime.

Then a clique-atom A of G is the closed neighbourhood of a non essential cut-vertex in H_{\max} if and only if it is a leaf in some atom tree of G . All the clique-atoms satisfying this above property can be computed in $\mathcal{O}(n + m)$ -time if an atom tree of G is given.

The above first assumption on the blocks may look a bit artificial. However, we emphasize that it holds for every regular graph [3]. Furthermore, we note that the condition of Theorem 11 is related to the notion of simplicial moplex (see [4]). In fact, the atoms satisfying the condition of the theorem are exactly the closed neighbourhoods of simplicial moplex.

Proof of Theorem 11. Let (T_G, \mathcal{A}) be an atom tree of G . Let $A \in \mathcal{A}$ be an arbitrary clique-atom. First note that since clique-cutsets are in correspondence with the edges in an atom tree, A is a leaf in some atom tree if and only if there exists a clique cutset $S \subseteq A$ such that, for every component C of $G \setminus A$, we have $N_G(C) \subseteq S$.

Suppose that $v \in A$ is a non essential cut-vertex of H_{\max} such that $N_{H_{\max}}[v] = A$. Observe that since G is not a complete graph, we have $V \neq A$. Furthermore, since v is non essential, there is a unique connected component D of $H_{\max} \setminus v$ such that $D \not\subseteq A$. So, let $S = (N_{H_{\max}}(v) \cap D) \cup \{v\}$. By construction, S is a clique and we have for every component C of $G \setminus A$, $N_G(C) \subseteq S$. We claim that S is a clique-cutset of G (i.e., a *minimal* separator). Indeed, in order to prove the claim we first need to observe that for every $u, w \in S$ non-adjacent in H_{\max} , there exist two uw -paths in H_{\max} that are internally vertex-disjoint (namely, (u, v, w) and any uw -path in D). Hence, there exists a block B of H_{\max} such that $S \subseteq V(B)$.

We prove as a subclaim that $V(B) \subseteq A'$ where $A' \neq A$ is an atom of G . There are two subcases to be distinguished.

- Suppose that B is a complete subgraph of H_{\max} . In this situation, $V(B) \subseteq A$ because $v \in S \subseteq V(B)$ and so, $V(B) = S$. Since $V(B) \setminus \{v\} \subseteq D \not\subseteq A$, there exists a cut-vertex v' of H_{\max} such that $v' \in V(B) \setminus \{v\}$ and so, $V(B) \subseteq A_{v'}$. So, we can choose $A' = A_{v'}$.
- Otherwise, B is not a complete subgraph of H_{\max} . By the hypothesis, B has no dominated vertex. Hence, $V(B)$ is an atom of G by Theorem 6, and so, we can choose $A' = V(B)$.

The latter proves the subclaim. Then, by Lemma 3, there exists a connected component C_S of $G \setminus A$ such that $A' \setminus S \subseteq C_S$ and $S \subseteq N_G(A' \setminus S) \subseteq N_G(C_S)$. It implies $N_G(C_S) = S$. Overall, this proves as claimed that S is a clique-cutset of G , and so, the clique-atom A indeed satisfies the condition of the theorem.

Conversely, suppose that A satisfies the condition of the theorem, and let us prove that it is the closed neighbourhood of a non essential cut-vertex of H_{\max} . Let $S \subseteq A$ be the clique cutset as defined by the condition of the theorem.

We first prove as an intermediary claim that A cannot be the closed neighbourhood of an essential cut-vertex. Indeed, since S is a clique cutset, and so, a minimal separator, there are at least two full components of $G \setminus S$. The latter implies the existence of a connected component C_S of $G \setminus A$ such that $N_G(C_S) = S$. However in this situation, let P, Q be any bipartition of the connected components of $G \setminus A$ such that $C_S \in P$. Since A satisfies the condition of the theorem, we have $N_G(Q) \subseteq N_G(P) = S$. Hence $N_G(P) \cap N_G(Q) = N_G(Q)$ cannot be reduced to a singleton because G is biconnected. By Lemma 8, this implies as claimed that A cannot be the closed neighbourhood of an essential cut-vertex.

Finally, suppose for the sake of contradiction that A is not the closed neighbourhood of a cut-vertex of H_{\max} . By Theorem 6, A is an atom in the square of some block of H_{\max} . By the hypothesis, it implies that A induces a block of H_{\max} , and this block cannot be a complete subgraph (otherwise, A would be contained in the closed neighbourhood of a cut-vertex in H_{\max}). Therefore, $H_A = H[A]$ has no dominated vertex. We prove as a new intermediary claim that A contains

a unique cut-vertex of H_{\max} . Indeed, the set of neighbourhoods of the connected components of $G \setminus A$ is exactly the set of intersections $A_v \cap A = N_{H_A}[v]$, $A_v = N_{H_{\max}}[v]$, for every cut-vertex v of H_{\max} contained in A . In particular, $A_v \cap A = N_{H_A}[v] \subseteq S$. Furthermore, recall that there exists a connected component C_S of $G \setminus A$ such that $N_G(C_S) = S$. Hence, there exists $v_0 \in C_S$ such that v_0 is a cut-vertex of H_{\max} and $A_{v_0} \cap A = N_{H_A}[v_0] = S$. Overall, since H_A has no dominated vertex, it implies as claimed that v_0 is the unique cut-vertex of H_{\max} contained in A . However, in this situation, let $v \in N_{H_A}(v_0)$ be arbitrary. Let H_0 be obtained from H_{\max} by transforming S into a clique and making every vertex of $A \setminus S$ a pending vertex adjacent to v . By construction, H_0 is a square root of G which is finer than H_{\max} . Since the blocks of H_0 still satisfy the two assumptions of the theorem, it contradicts the minimality of H_{\max} . As a result, we obtain that A is the closed neighbourhood of some cut-vertex of H_{\max} (necessarily, non essential).

In order to complete the proof, we are left to prove the linear time bound in order to compute all the clique-atoms that satisfy the desired property. We recall that given (T_G, \mathcal{A}) , all the clique-atoms can be computed in $\mathcal{O}(n+m)$ -time [15]. Furthermore, for every clique-atom A , let $\Omega^*(A) = \{A' \cap A \mid \{A', A\} \in E(T_G)\}$ and let $S_A \in \Omega^*(A)$ be of maximum size. By the combination of Lemma 3 with usual properties of tree decompositions, we have that if A satisfies the condition of the theorem then it does so with $S = S_A$. It can be checked in time $\mathcal{O}(\sum_{S^* \in \Omega^*(A)} |S^*|)$. Overall, all the clique-atoms that satisfy the condition of the theorem can be retrieved in time:

$$\mathcal{O}\left(\sum_{A \in \mathcal{A}} \sum_{S^* \in \Omega^*(A)} |S^*|\right) = \mathcal{O}\left(\sum_{\{A, A'\} \in E(T_G)} |A \cap A'|\right),$$

that is $\mathcal{O}(n+m)$ by Lemma 4. □

5. Reconstructing the block-cut tree of a square root

Given a graph $G = (V, E)$, we propose a generic approach in order to compute the block-cut tree of one of its square-roots (if any). More precisely, we remind that a square root H_{\max} of G is called a *maxblock square root* if there does not exist any other square root $H \neq H_{\max}$ of G with all its blocks being contained in the blocks of H_{\max} . We suppose we are given the closed neighbourhoods of all the cut-vertices in some maxblock square root H_{\max} of G (the cut-vertices may not be part of the input). Based on this information, we show in Section 5.1 how to compute for every block of H_{\max} a graph that is isomorphic to its square.

However, in doing so, the correspondence between the nodes in these graphs and the nodes in G is partly lost. Hence for each block, we need to solve a stronger version of the \mathcal{H} -SQUARE ROOT problem in order to obtain a global solution for G . This new version of the problem is introduced in Section 5.2.

Finally, in Section 5.3, we sketch a few requirements on the class \mathcal{H} in order for our framework to be applicable in a somewhat automatic way.

5.1. Computing the squares of the blocks

This section is devoted to the proof of the following theorem:

Theorem 12. *Let H_{\max} be a maxblock square root of a graph $G = (V, E)$, and let A_1, A_2, \dots, A_k be the closed neighbourhoods of every cut-vertex in H_{\max} . For every block B of H_{\max} , we can compute*

a graph G_B that is isomorphic to its square. Furthermore, if B is not isomorphic to K_2 then we always have $G_B = B^2$. It can be done in $\mathcal{O}(n + m)$ -time in total if an atom tree of G is given.

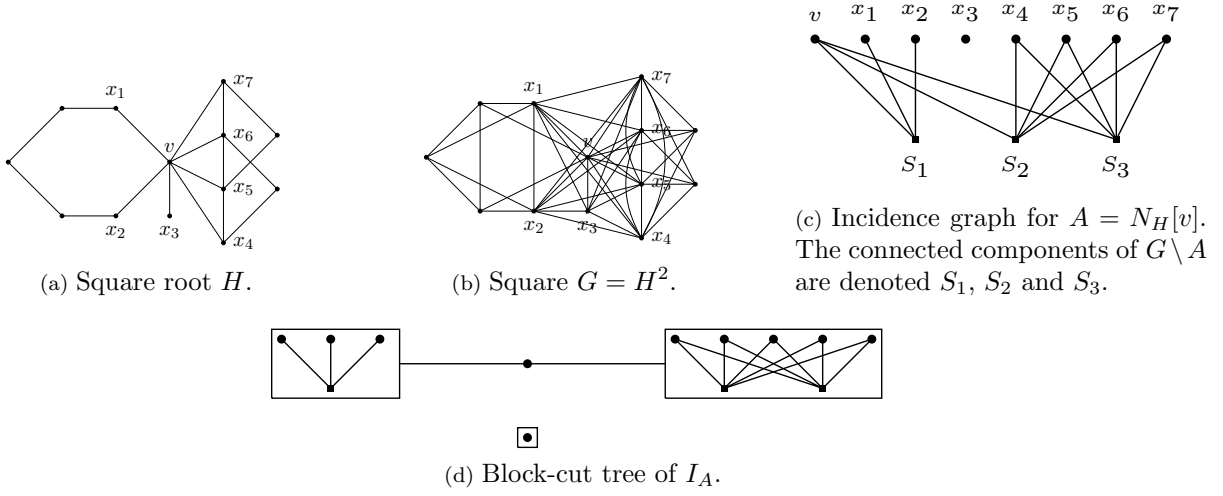


Figure 7: Computation of the connected components in a square root.

Our proof is based on the incidence graphs presented in Section 4.1. More precisely, we recall that for every clique-atom A of G , we define $\Omega(A)$ as the multiset containing $N_G(C)$ for every connected component C of $G \setminus A$. The incidence graph $I_A = Inc(\Omega(A), A)$ is the bipartite graph with respective sides $\Omega(A)$ and A and an edge between every $S \in \Omega(A)$ and every $u \in S$.

We need the following technical lemmas.

Lemma 13. *Let H_{\max} be a maxblock square root of a graph $G = (V, E)$. A vertex has a maximum neighbour in H_{\max} if and only if it is a pending vertex.*

Proof. On the one hand, let $v \in V$ be a pending vertex of H_{\max} with u being its unique neighbour in the square root. Clearly, u is a maximum neighbour of v in H_{\max} . Conversely, let $u, v \in V$ be such that u is a maximum neighbour of v in H_{\max} . We have that $N_G[v] = N_{H_{\max}}[u]$. In particular, let H_0 obtained from $H_{\max} \setminus v$ by adding vertex v and the edge $\{u, v\}$. Since u is a maximum neighbour of v , we have that H_0 keeps the property to be a square root of G . Furthermore, since H_{\max} is a maxblock square root, we obtain that $H_0 = H_{\max}$. Hence, v is a pending vertex. \square

Given a clique-atom A and its incidence graph I_A , we can compute the blocks of I_A . Then, let us define the following equivalence relation over the connected components of $G \setminus A$: $C \sim_A C'$ if and only if $N_G(C)$ and $N_G(C')$ (taken as elements of $\Omega(A)$) are in the same block of I_A . The latter relation naturally extends to an equivalence relation over $V \setminus A$: for every components C, C' of $G \setminus A$ and for every $u \in C, u' \in C', u \equiv_A u'$ if and only if $C \sim_A C'$. In doing so, the equivalence classes of \equiv_A partition the set $V \setminus A$. We refer to Figure 7 for an illustration of the procedure.

Lemma 14. *Let $G = (V, E)$ be a graph, let H_{\max} be a maxblock square root of G , and let A be a clique-atom of G . Suppose that A is the closed neighbourhood of a cut-vertex v in H_{\max} and let C_1, C_2, \dots, C_ℓ be the nontrivial connected components of $H_{\max} \setminus v$. Then the equivalence classes of \equiv_A are exactly the sets $C_i \setminus A, 1 \leq i \leq \ell$.*

Proof. The proof follows from a more careful use of the arguments in the proofs of Lemmas 8 and 9 (that we will repeat for the convenience of the reader). On the one direction, we claim that each equivalence class of \equiv_A intersects exactly one of the sets $C_i \setminus A$. In order to prove the claim, we first observe that this holds for the *non essential* cut-vertices (trivially). Thus from now on, let us assume that v is an essential cut-vertex of H_{\max} . Let D be a connected component of $G \setminus A$. As observed in the proof of Lemma 8, for every $i \neq j$ and for every $u_i \in C_i \setminus A_v$, $u_j \in C_j \setminus A_v$, we have $\text{dist}_H(u_i, u_j) = \text{dist}_H(u_i, v) + \text{dist}_H(u_j, v) \geq 4$. Therefore, D intersects exactly one of the sets $C_i \setminus A$. Furthermore for every $i \neq j$ we have that if $D \subseteq C_i \setminus A$ and $D' \subseteq C_j \setminus A$ then, $N_G(D) \cap N_G(D') \subseteq \{v\}$, that implies $N_G(D)$ and $N_G(D')$ are in different connected components of $I_A \setminus v$. Hence, each equivalence class of \equiv_A intersects exactly one of the sets $C_i \setminus A$.

Conversely, let i be fixed, $1 \leq i \leq \ell$, and let $D, D' \subseteq C_i \setminus A$ be two connected components of $G \setminus A$. First note that since $N_{H_{\max}}(D) \cup N_{H_{\max}}(D') \subseteq A = N_{H_{\max}}[v]$, therefore $v \in N_G(D) \cap N_G(D')$. In addition, and as noted in the proof of Lemma 9, if there exists a bipartition of the connected components of $G \setminus A$, in P and Q say, such that $N_G(P) \cap N_G(Q) = \{v\}$, then the vertices in P are in different components of $H_{\max} \setminus v$ than the vertices in Q (otherwise, we could transform H_{\max} so that it is the case and, in doing so, we would increase the number of blocks). In particular, for any such a bipartition we must have either $D, D' \in P$ or $D, D' \in Q$. Consequently, $N_G(D)$ and $N_G(D')$ (taken as vertices of I_A) cannot be disconnected by v . It implies the existence of two internally vertex-disjoint $N_G(D)N_G(D')$ -paths of I_A (namely, $(N_G(D), v, N_G(D'))$ and any $N_G(D)N_G(D')$ -path in $I_A \setminus v$. Altogether combined, $N_G(D), N_G(D') \in \Omega(A)$ are in a common block of I_A . In particular, all the components that intersect the set $C_i \setminus A$ are in the same equivalence class of \equiv_A . \square

We are now ready to prove Theorem 12.

Proof of Theorem 12. Our approach mimics the following "naive" algorithm for computing the blocks of H_{\max} . Let $v \in V$ be an arbitrary cut-vertex. We compute the connected components C_1, \dots, C_ℓ of $H_{\max} \setminus v$. Then, it follows from the existence of the block-cut tree that the blocks of H_{\max} are exactly the blocks contained in the subgraphs $H_{\max}[C_1 \cup \{v\}], \dots, H_{\max}[C_\ell \cup \{v\}]$. Each of the subgraphs can thus be considered separately.

More formally, we consider the cut-vertices v_1, \dots, v_k of H_{\max} sequentially. At each step i we maintain a T -decomposition of H_{\max} , that is, a collection of subgraphs \mathcal{T}_i with the property that every block is contained in a unique subgraph of the collection. Initially, $\mathcal{T}_1 = \{H_{\max}\}$. We ensure that each cut-vertex v_i, v_{i+1}, \dots, v_k is contained in a unique subgraph of \mathcal{T}_i . In particular, we consider the unique subgraph H_i that contains v_i and we compute the components C_1, \dots, C_ℓ of $H_i \setminus v_i$. We finally construct the collection \mathcal{T}_{i+1} from $\mathcal{T}_i \setminus \{H_i\}$ by adding, for every $1 \leq j \leq \ell$, the subgraph $H_i[C_j \cup \{v_i\}]$ to the current collection. Note that the graphs in the final set \mathcal{T}_{k+1} are exactly the blocks of H_{\max} .

Back to the square G , we consider the sets A_1, A_2, \dots, A_k sequentially. Our purpose for proving Theorem 12 is to maintain an "atom forest" \mathcal{F}_i : with an atom tree for the square of every subgraph in \mathcal{T}_i . This cannot be done in general, as we may not know the cut-vertices of H_{\max} (we know for sure their closed neighbourhoods). However, we prove that it can be done for all the subgraphs of \mathcal{T}_i that are not reduced to a pending vertex and its (unique) neighbour in H_{\max} .

We set initially $\mathcal{F}_1 = \{(T_G, \mathcal{A})\}$ with (T_G, \mathcal{A}) being an atom tree of G . Furthermore, at each step i we ensure that each set A_i, A_{i+1}, \dots, A_k is a bag in a unique tree decomposition in \mathcal{F}_i . For

every $1 \leq i \leq k$, we consider the unique $(T_i, \mathcal{A}_i) \in \mathcal{F}_i$ such that $A_i \in \mathcal{A}_i$. It somewhat represents an atom tree of $G_i = H_i^2$. Let $\Omega^*(A_i) = \{A_i \cap A' \mid \{A_i, A'\} \in E(T_G)\}$.

- First we compute the number s_i of vertices of G_i that are uniquely contained in A_i . It can be computed in time $\mathcal{O}(|A_i| + \sum_{S^* \in \Omega^*(A_i)} |S^*|)$. By Lemma 2, all these vertices are simplicial in G , with their closed neighbourhood in G being A_i . Since v_i is a maximum neighbour (in the square root) for all the simplicial vertices in A_i , by Lemma 13 there are exactly s_i trivial connected components of $H_{\max} \setminus v_i$. Note that each such component induces a block isomorphic to K_2 .
- We then compute the blocks of the incidence graph $I_{A_i} = \text{Inc}(\Omega(A_i), A_i)$. Note that each block represents an equivalence class of \equiv_{A_i} in G (not in G_i). Furthermore, as explained in the proof of Theorem 7, it can be done using the reduced incidence graph $I_{A_i}^* = \text{Inc}(\Omega^*(A_i), A_i)$. So, it can be done in time $\mathcal{O}(\sum_{S^* \in \Omega^*(A_i)} |S^*|)$.

Let C_1, C_2, \dots, C_q be the remaining $q = \ell - s_i$ connected components of $H_i \setminus v$ (we remove from the list the trivial components that correspond to the s_i simplicial vertices in A_i). There is a bijection between these components and the nontrivial components C'_1, C'_2, \dots, C'_q of $H_{\max} \setminus v$. Moreover by Lemma 14, the set of equivalence classes of \equiv_{A_i} is exactly $\{C'_j \setminus A_i \mid 1 \leq j \leq q\}$. Since by Lemma 13 vertex v_i cannot be a maximum neighbour of the vertices in these components, we can thus obtain the sets $C_j \cup \{v_i\}$ from $N_G[C'_j \setminus A_i] \cap V(H_i)$, for every $1 \leq i \leq q$.

- By the properties of tree decompositions, each set $C_j \cup \{v\}$ corresponds to a collection \mathcal{C}_j of the subtrees in $T_i \setminus A_i$. Namely, these subtrees are obtained from the blocks of $I_{A_i}^*$, by grouping the subtrees whose respective intersections with A_i are in the same block of $I_{A_i}^*$. Let us remove all the subtrees of $T_i \setminus A_i$ that are not part of \mathcal{C}_j . Then, let us replace A_i by $A_i \cap (C_j \cup \{v\})$. We so obtain a tree decomposition $(T_{i,j}, \mathcal{A}_{i,j})$ for the square of $H_i[C_j \cup \{v\}]$. Finally, in order to obtain \mathcal{F}_{i+1} from $\mathcal{F}_i \setminus \{(T_i, \mathcal{A}_i)\}$, we add all the $(T_{i,j}, \mathcal{A}_{i,j})$ in the collection, for every $1 \leq j \leq q$.

Formally, the tree decompositions $(T_{i,j}, \mathcal{A}_{i,j})$ are not exactly atom trees since they may contain non maximal prime subgraphs (*e.g.*, the bag $A_i \cap (C_j \cup \{v\})$). However, it is necessary to keep these additional bags for the correctness of the algorithm. Indeed, consider the special case of a block B that is dominated by at least two cut-vertices v_{i_1}, v_{i_2} . At the i_1^{th} step, we replace A_{i_1} by $V(B)$ in the unique tree-decomposition $(T_{i_1,j}, \mathcal{A}_j)$ containing A_{i_2} . This is clearly not an atom since $V(B) \subseteq A_{i_2}$. However, deleting this bag could result in missing B . This is also why we need to compute \equiv_{A_i} in G rather than in the subgraph G_i .

Overall, the time complexity is in:

$$\mathcal{O}\left(\sum_{A \in \mathcal{A}} \sum_{S^* \in \Omega^*(A)} |S^*|\right) = \mathcal{O}\left(\sum_{\{A, A'\} \in E(T_G)} |A \cap A'|\right),$$

that is $\mathcal{O}(n + m)$ by Lemma 4.

□

Having a closer look at our reconstruction method, it computes for *any* square root (not necessarily maxblock) a collection of subsets of its blocks. We need Lemmas 13 and 14 in order to ensure that no subset can be proper, *i.e.*, the subsets computed are exactly the blocks of the root.

5.2. Obtaining a global solution from the blocks

Then, we wish to solve the \mathcal{H} -SQUARE ROOT problem for each square of a block separately. However, doing so, we may not be able to reconstruct a square root for the original graph. Indeed, the closed neighbourhoods of cut-vertices are imposed, and these additional constraints may be violated by the partial solutions. We thus need to solve the following stronger version of the problem.

Problem 2 (\mathcal{H} -SQUARE ROOT WITH NEIGHBOURS CONSTRAINTS).

Input: A graph $G = (V, E)$; a list \mathcal{N}_F of pairs $\langle v, N_v \rangle$ with $v \in V$, $N_v \subseteq V$; a list \mathcal{N}_A of subsets $N_i \subseteq V$, $1 \leq i \leq k$.

Question: Are there a graph $H \in \mathcal{H}$ and a collection $v_1, v_2, \dots, v_k \in V$ of pairwise different vertices such that H is a square root of G , and:

- $\forall \langle v, N_v \rangle \in \mathcal{N}_F$, we have $N_H[v] = N_v$
- $\forall 1 \leq i \leq k$, we have $N_H[v_i] = N_i$; furthermore, $\langle v_i, N_i \rangle \notin \mathcal{N}_F$?

To our best knowledge, this variant has not been studied before in the literature. We show how to solve it for some graph classes in the next section. Intuitively, the list \mathcal{N}_F represents the *essential* cut-vertices and their closed neighbourhoods in the block. The list \mathcal{N}_A represents the closed neighbourhoods of non essential cut-vertices. Furthermore, non essential cut-vertices correspond to the vertices v_1, \dots, v_k to be computed. Notice that we need to ensure that all the v_i 's are pairwise different in case there may be true twins in a square root. We also need to ensure that $\langle v_i, N_i \rangle \notin \mathcal{N}_F$ for the same reason.

5.3. Discussion

So far, most of our results have been proved for maxblock square roots. It may be difficult to show that the target class \mathcal{H} contains such a root. However, on a positive side, our results can sometimes be applied to a *finest* square root in \mathcal{H} . Before concluding this section, we sketch a few requirements on \mathcal{H} so that it can be done.

First we need that, for every graph in \mathcal{H} , its blocks satisfy the technical assumptions of Theorem 11. In particular, this is the case if each such a block has a prime square, and it is a regular graph. Interestingly, since every vertex with a maximum neighbour is dominated, Lemma 13 also holds in this case. This is worth noticing since the latter is one of the two main ingredients in our correctness proof of the reconstruction of the blocks.

Then, we must further restrict the properties of the blocks in order to solve the \mathcal{H}' -SQUARE ROOT WITH NEIGHBOURS CONSTRAINTS problem, with \mathcal{H}' being the graph class of all the blocks. For instance, we may impose that every block that is not a clique has a prime square of *bounded treewidth*. We note that the latter is always satisfied when every block that is not a clique has bounded treewidth and bounded maximum degree. Our motivation here is that the constraints of \mathcal{H}' -SQUARE ROOT WITH NEIGHBOURS CONSTRAINTS can be rewritten as an MSO_2 formula. More precisely, we prove the following lemma:

Lemma 15. *Suppose that the recognition of a graph in \mathcal{H}' can be written as an MSO_2 formula, and that every square of a graph in \mathcal{H}' has bounded treewidth. Then, we can solve \mathcal{H}' -SQUARE ROOT WITH NEIGHBOURS CONSTRAINTS in linear-time.*

Proof. Let us consider any instance $G; \mathcal{N}_F; \mathcal{N}_A$ of the problem. For every $N_i \in \mathcal{N}_A$ we add a fresh new vertex v_i that is made adjacent to every vertex of N_i . Since all the v_i 's are simplicial, the treewidth of G and of the resulting graph G' can differ by at most 1 (see [7]). Then, we can define the following sets:

$$V_A = \{v_i \mid N_i \in \mathcal{N}_A\}, \quad V_F = \{v \mid \langle v, N_v \rangle \in \mathcal{N}_F\}, \quad E_F = \{\{v, u\} \mid \langle v, N_v \rangle \in \mathcal{N}_F \text{ and } u \in N_v\}.$$

Assuming that the recognition of a graph in \mathcal{H}' can be written as an MSO_2 formula, we can encode \mathcal{H}' -SQUARE ROOT WITH NEIGHBOURS CONSTRAINTS by the following formula:

$$\begin{aligned} \exists E_H \subseteq E(G') \text{ s.t. } G'[E_H]^2 = G' \text{ and} \\ H = (V, E \cap E_H) \in \mathcal{H}', \\ \forall v \in V_F, \forall u \in V(G'), \{v, u\} \in E_H \iff \{v, u\} \in E_F, \\ \forall v_i \in V_A, \exists! u_i \in V(G') \text{ s.t. } \{v_i, u_i\} \in E_H, \\ u_i \notin V_F \cup V_A, \\ \forall v_j \in V_A \setminus \{v_i\}, u_i \neq u_j. \end{aligned}$$

The fourth constraint ensures that all the newly added v_i 's are pending vertices in the root. In particular, this implies that H (as defined in the second constraint) is a root of the original graph G . Furthermore, the third constraint ensures that for every $\langle v, N_v \rangle \in \mathcal{N}_F$, we have $N_H[v] = N_v$. Since for every $v_i \in V_A$ we have by construction $N_{G'}(v_i) = N_i$, the fourth constraint also implies the existence of vertices u_1, u_2, \dots, u_k with respective closed neighbourhoods in the root being N_1, N_2, \dots, N_k . The fifth constraint ensures that all the u_i 's are vertices of the original G , and the sixth constraint ensures that they are pairwise different. Finally, the fifth constraint also ensures that $\langle u_i, N_i \rangle \notin \mathcal{N}_F$.

Overall, if G (and so, G') has bounded treewidth, we can apply Courcelle's theorem [16] to the above formula in order to solve \mathcal{H}' -SQUARE ROOT WITH NEIGHBOURS CONSTRAINTS in linear-time. \square

We stress that the approach taken in Lemma 15 has been followed in other papers, for another variant of the problem called \mathcal{H}' -SQUARE ROOT WITH LABELS [23].

Last, in order for the framework to be applied, we need to compute the essential cut-vertices of some finest square root in \mathcal{H} . We did not find a satisfying property for that, as the graph operations that we use in the proof of Lemma 9 (for proving the correctness of Algorithm 1) are not very natural. Lemma 19 in the next section suggests that the existence of a suitable (Helly) H -representation of the blocks could be helpful in order to prove the correctness of Algorithm 1 for more interesting cases. The algorithmic uses of such representations already have attracted some attention in the literature (see [13]).

6. Application to trees of cycle-powers

A cycle-power graph is any j^{th} -power C_n^j of the n -node cycle C_n , for some $j, n \geq 1$. A *tree of cycle-powers* is a graph whose blocks are cycle-power graphs. In particular, a j -cactus-block graph

is a graph whose blocks are complete graphs or k^{th} -powers of cycles, for any $1 \leq k \leq j$. This above class generalizes the classes of trees, block graphs and cacti: where all the blocks are edges, complete subgraphs and cycles, respectively. Other relevant examples are the class of *cactus-block graphs* (a.k.a., 1-cactus-block graphs with our terminology): where all the blocks are either cycles or complete subgraphs [47]; and the *Gallai trees*, that are the cactus-block graphs with no block being isomorphic to an even cycle [22]. Our main result in this section is that the squares of these graphs can be recognized in polynomial time:

Theorem 16. *For every fixed $j \geq 1$, the squares of j -cactus-block graphs can be recognized in $\mathcal{O}(nm)$ -time.*

Up to simple changes, the proof of Theorem 16 applies to all the subclasses mentioned above. This solves for the first time the complexity of the \mathcal{H} -SQUARE ROOT problem for the cactus-block graphs and Gallai trees:

Theorem 17. *Squares of cactus-block graphs, resp. squares of Gallai trees, can be recognized in $\mathcal{O}(nm)$ -time.*

On the complexity point of view, our algorithm runs in $\mathcal{O}(nm)$ -time. This improves upon the $\mathcal{O}(n^4)$ -time algorithm of [24] for the recognition of squares of cacti. Furthermore, the main bottleneck is the computation of an atom tree for the graph, that requires $\mathcal{O}(nm)$ -time [5]. Indeed, all the other operations can be performed in linear-time. It implies that for graphs where an atom tree is easier to compute, such as chordal squares, we achieve a better time complexity. In particular, we retrieve as particular cases the linear-time algorithms for the recognition of squares of trees and block graphs of [33].

The proof of Theorem 16 is twofold. We seek for a square root H of G that is a tree of cycle-powers and maximizes its number of blocks. First we prove in Lemma 19 that the cut-vertices in this square root are exactly those characterized by Theorems 7 and 11. We do so by adapting the respective techniques from Lemma 9 and Theorem 11 in order to increase the number of cut-vertices. An illustration is provided with Figure 8. Then, we need to show that \mathcal{H} -SQUARE ROOT WITH NEIGHBOURS CONSTRAINTS can be solved in linear time for j -cactus-block graphs (Lemma 20). This is done by exploiting the fact that cycle-power graphs are *circular-arc graphs* (intersection graphs of intervals on the cycle) with a unique circular-arc model [26, 36].

6.1. Existence of a nice square root

We start with a basic property of cycle-power graphs.

Lemma 18. *For every $j, n \geq 1$, the cycle-power graph C_n^j is prime.*

Proof. This is clear if $j \geq \lfloor n/2 \rfloor$ for then C_n^j is isomorphic to the complete graph K_n . Thus from now on, assume $1 \leq j < \lfloor n/2 \rfloor$. Let us label the vertices of C_n^j by \mathbb{Z}_n , in such a way that vertex i is adjacent to the vertices $i \pm k$, $1 \leq k \leq j$. It has been noted in [17] that every minimal separator S of C_n^j is the union of two disjoint "circular intervals", i.e., $S = \{i_1, i_1 + 1, \dots, i_1 + j - 1\} \cup \{i_2, i_2 + 1, \dots, i_2 + j - 1\}$ for some i_1, i_2 . Furthermore, the only two (full) components of $C_n^j \setminus S$ have respective vertex-sets $\{i_1 + j, i_1 + j + 1, \dots, i_2 - 1\}$ and $\{i_2 + j, i_2 + j + 1, \dots, i_1 - 1\}$ (indices are taken modulo n). In this situation, i_1 and i_2 are at distance at least $j + 1$ in C_n . So, they cannot be adjacent in C_n^j . As a result, no minimal separator of C_n^j can be a clique, hence C_n^j is prime. \square

In order to prove the next result, we use classical techniques in the study of circular-arc graphs (intersection graphs of intervals on the cycle) [52]. Indeed, every cycle-power graph is a circular-arc graph [36].

Lemma 19. *Let $G = (V, E)$ be a graph that is not a complete graph. Let H_{\max} be a finest square root of G that is a j -cactus-block graph. The set of closed neighbourhoods of the cut-vertices in H_{\max} can be computed in $\mathcal{O}(n + m)$ -time if an atom tree of G is given.*

Proof. Let H be any square root of G that is a j -cactus-block graph. First we observe that every cycle-power graph is regular. Furthermore, by Lemma 18 every cycle-power graph has a prime square. Hence, H satisfies the two assumptions of Theorem 11. In the proof of Theorem 11, we propose a graph modification in order to increase the number of non essential cut-vertices. This operation keeps the property for the root to be a j -cactus-block graph (see Fig. 8a and 8b for an illustration.). Therefore, by maximality of the number of blocks in H_{\max} , the closed neighbourhoods of non essential cut-vertices in H_{\max} are exactly the clique-atoms that satisfy the condition of Theorem 11.

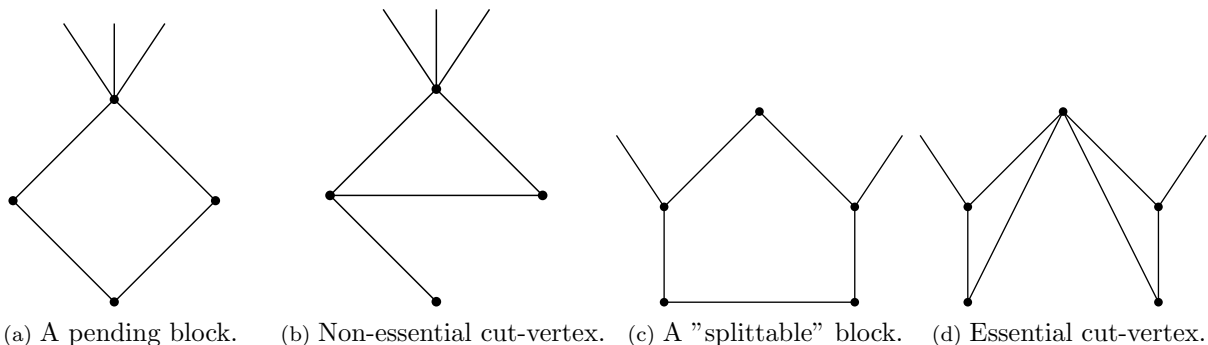


Figure 8: Local modifications of the blocks.

Similarly, let us consider any clique-atom of G passing the test of Algorithm 1. If it is not the closed neighbourhood of an essential cut-vertex then it corresponds to a block B of H with diameter two. Thus, $B = C_n^k$, $\lceil [n/2] / 2 \rceil \leq k \leq \min\{j, [n/2] - 1\}$. Let v_1, v_2, \dots, v_c , $c \geq 2$ be the cut-vertices that are contained in $V(B)$. Recall that all the sets $N_{H_{\max}}[v_1], N_{H_{\max}}[v_2], \dots, N_{H_{\max}}[v_c]$ are atoms intersecting B . Furthermore for any two i, i' the block B is the only one that contains both $v_i, v_{i'}$ (*i.e.*, due to the existence of the block-cut tree). Hence, the sets $N_{H_{\max}}[v_i] \setminus V(B)$ are in different connected components of $G \setminus B$. Conversely, every connected component of $G \setminus B$ must contain exactly one of these above sets.

By the choice of B , the connected components of $G \setminus B$ can be bipartitioned into two nonempty parts P, Q such that $N_G(P) \cap N_G(Q) = \{v\}$. Note that for the component that contains $N_{H_{\max}}[v_i] \setminus V(B)$, its neighbourhood in G is $N_{C_n^k}[v_i]$. Say that such a neighbourhood spans an edge if it contains its two ends. We claim that the neighbourhoods $N_{C_n^k}[v_i]$ of these above components cannot span all the edges of C_n (the underlying cycle of the block). Indeed, otherwise any bipartition of these components into two non empty parts P and Q would result in $|N_G(P) \cap N_G(Q)| \geq 2$. In this situation, every set $N_{C_n^k}[v_i]$ can be represented as a subpath of size $2k + 1$ on a path P_n with n elements (*i.e.*, we transform C_n into a path by removing an edge not spanned). Furthermore, the sets $N_{C_n^k}[v_i]$ are pairwise intersecting since B is a complete subgraph of G . Thus, by the Helly

property, $\bigcap_{i=1}^c N_{C_n^k}[v_i] \neq \emptyset$. It implies $\bigcap_{i=1}^c N_{C_n^k}[v_i] = \{v\}$ since otherwise, there could not exist a bipartition P, Q of the connected components of $G \setminus B$ such that $N_G(P) \cap N_G(Q) = \{v\}$. For the same reason as above, there can be no neighbourhood $N_{H_{\max}}[v_i]$ that spans the at most two edges of P_n incident to v . It implies that v is an internal node of P_n and every neighbourhood $N_{H_{\max}}[v_i]$ must be fully contained into one of the two maximal subpaths of P_n of which v is an end. Finally, since all the sets $N_{C_n^k}[v_i]$ have equal size $2k + 1$, the only possibility so that $\bigcap_{i=1}^c N_{C_n^k}[v_i] = \{v\}$ is that $c = 2$.

Our construction to increase the number of essential cut-vertices is now as follows. We split the underlying cycle C_n of $B = C_n^k$ into the three intervals $I_1 = N_{C_n^k}[v_1] = \{i - k, i - k + 1, \dots, i = v_1, i + 1, \dots, i + k = v\}$, $I_2 = N_{C_n^k}[v_2] = \{i + k = v, i + k + 1, \dots, i + 2k = v_2, i + 2k + 1, \dots, i + 3k\}$ and $I_3 = B \setminus (I_1 \cup I_2)$. By construction, $I_1 \cap I_2 = \{v\}$ and $I_1 \cap I_3 = I_2 \cap I_3 = \emptyset$. Then, we replace C_n^k by two cliques with respective vertex-sets I_1, I_2 (both isomorphic to C_{2k+1}^k) and pending vertices $i_3 \in I_3$ only adjacent to v . See Fig. 8c and 8d for an illustration. This operation preserves the property for the square root to be a j -cactus-block graph. Therefore, by maximality of the number of blocks in H_{\max} , the essential cut-vertices of H_{\max} and their closed neighbourhood are exactly those computed by Algorithm 1. \square

6.2. Solving a problem on circular intervals

Our proof of the previous Lemma 19 uses classical techniques in the study of circular-arc graphs (intersection graphs of intervals on the cycle) [52]. A circular-arc model is a mapping between the nodes of a circular-arc graph and some arcs/intervals on a cycle so that two vertices are adjacent if and only if the corresponding arcs intersect. We further exploit the relationship with this class of graphs and cycle-power graphs in order to prove Lemma 20.

Lemma 20. *Let $j \geq 1$ be a fixed constant. \mathcal{H} -SQUARE ROOT WITH NEIGHBOURS CONSTRAINTS can be solved in linear-time when \mathcal{H} is the class of j -cactus-block graphs.*

Proof. Let $G = (V, E)$; \mathcal{N}_F ; \mathcal{N}_A be an instance of the problem. There are two cases to be considered.

- Suppose that G is a complete graph. If there is no further constraint (*i.e.*, $\mathcal{N}_A = \mathcal{N}_F = \emptyset$) then G ; \mathcal{N}_F ; \mathcal{N}_A is trivially a yes-instance. This is also the case if for every $\langle v, N_v \rangle \in \mathcal{N}_F$ we have $|N_v| = n$, and similarly for every $N_i \in \mathcal{N}_A$ we have $|N_i| = n$. Let us now assume that all these sets are pairwise different and they have equal odd size $2k + 1$ (supposedly representing the closed neighbourhood of some vertex in C_n^k) for some $1 \leq k \leq j$. In this situation, since G is a complete graph we have $n \leq 4k + 1 \leq 4j + 1$. Furthermore since j is assumed to be a constant by the hypothesis, it implies that the graph G has bounded size, and so, the \mathcal{H} -SQUARE ROOT WITH NEIGHBOURS CONSTRAINTS can be solved in constant-time by brute force⁶.

⁶Note, perhaps surprisingly, that this first case is the most difficult to solve with our methods. Indeed, solving the \mathcal{H} -SQUARE ROOT WITH NEIGHBOURS CONSTRAINTS when G is a complete graph and j is arbitrary would directly lead to a polynomial-time recognition algorithm for the squares of trees of cycle-powers.

- Else, G is not a complete graph. Since G is assumed to be the square of a cycle-power graph, there must exist some $1 \leq k \leq j$ such that: $k < \lceil [n/2] / 2 \rceil$; G is isomorphic to C_n^{2k} ; the sets N_v , for every $\langle v, N_v \rangle \in \mathcal{N}_F$, and the sets $N_i \in \mathcal{N}_A$, are pairwise different and they have an equal odd size $2k + 1$ (supposedly representing the closed neighbourhood of some vertex in C_n^k). It can be verified in linear-time [36]. Furthermore, in such situation, we are left for proving the existence of a one-to-one mapping $\varphi : V \rightarrow \mathbb{Z}_n$ such that: φ is an isomorphism of G and C_n^{2k} ; for every $\langle v, N_v \rangle \in \mathcal{N}_F$, the vertices in N_v are consecutive with v being in the middle of the interval; each set $N_i \in \mathcal{N}_A$, is a "circular interval" of consecutive points on the cycle C_n .

Given φ , we can construct a circular-arc model for $G = C_n^{2k}$ in the natural way with $\varphi(V) = \mathbb{Z}_n$ representing the extremal points of the arcs. It follows from [26] that a cycle-power graph that is not a complete graph has a unique such model up to rotation and reflection. So, let us compute this model. It takes linear-time [39]. Moreover, as proved in [26], we can obtain the desired mapping φ by labeling the vertices of G following a circular ordering of the starting points in this model. In such a situation, we are left to check whether all the neighbours constraints of \mathcal{N}_F and \mathcal{N}_A are satisfied by φ . It can be done in linear-time.

□

Altogether combined, we can finally prove Theorem 16.

Proof of Theorem 16. Given a graph $G = (V, E)$, the problem is trivial when G is a complete graph. So, we assume from now on that G is not a complete graph. We compute an atom tree of G . It can be done in $\mathcal{O}(nm)$ -time [5]. Our objective is to compute a finest square root H of G that is a j -cactus-block graph. Assuming H exists, by Lemma 19 all the closed neighbourhood of the cut-vertices in H can be computed in $\mathcal{O}(n + m)$ -time.

Then, we aim at computing the squares of the blocks of H , using Theorem 12. For this purpose, we only need to prove that the results of Lemmas 13 and 14 hold for H . First we claim that Lemma 13 holds for H .

Indeed, observe that since in any cycle-power graph that is not a complete graph, there cannot be a dominated vertex, therefore there cannot be any vertex with a maximum neighbour either. It follows from this observation that, provided H exists, a vertex v of H can be a maximum neighbour of some other vertex w only if v and w are contained in a block B of H that induces a complete subgraph. Furthermore, v is the only possible cut-vertex contained in B (otherwise, v could not possibly be a maximum neighbour of w). In such situation, the clique B can be "splitted" into pending vertices adjacent to v . Hence, as claimed, the result of Lemma 13 holds for any *finest* square root H that is a j -cactus-block graph.

Then, we claim that Lemma 14 also holds for H . For proving the claim, let A be the closed neighbourhood of a cut-vertex v in H . As noted in Lemma 8 we have that every equivalence class of \equiv_A intersects a unique connected component of $H \setminus v$. Conversely, we claim that for every nontrivial component C_i of $H \setminus v$, the subset $C_i \setminus A$ is contained in a unique equivalence class of \equiv_A . Indeed, either $C_i \setminus A$ is a connected subset of G (and so, we are done); or then, since by Lemma 18 the squares of the blocks of H are prime, there must be two blocks $B_1, B_2 \subseteq C_i \cup \{v\}$ such that $B_1 \setminus A$ and $B_2 \setminus A$ are in different connected components of $G \setminus A$. In the latter case, let $B \subseteq C_i \cup \{v\}$ be the block of H containing $A \cap (C_i \cup \{v\})$ (possibly, $B = B_1$ or $B = B_2$). Furthermore, for every $i \in \{1, 2\}$, let D_i be the connected component of $G \setminus A$ containing $B_i \setminus A$.

Since we have by Lemma 3 that $B \setminus A$ is a connected subset of G (possibly empty), we can assume w.l.o.g. that $D_1 \cap (B \setminus A) = \emptyset$. Then, since both B, D_1 intersect C_i , there must be a cut-vertex $v' \neq v$ of H such that: $N_G(D_1) = N_H[v'] \cap B \subseteq A \cap (C_i \cup \{v\}) = N_H[v] \cap B$. In particular, B is a clique of H since $H[B]$ contains a dominated vertex, and so, $B = A \cap (C_i \cup \{v\})$. As a result, all the connected components of $G[C_i \setminus A]$ are adjacent to every vertex of B (since they share with B a cut-vertex of H), hence $C_i \setminus A$ is in a unique equivalence class of \equiv_A . So, the claim is proved.

Altogether combined, the latter is enough so that we can reuse the algorithm of Theorem 12 in order to compute, for every block of H , a graph isomorphic to its square. It can be done in $\mathcal{O}(n + m)$ -time. We finally need to solve the \mathcal{H} -SQUARE ROOT WITH NEIGHBOURS CONSTRAINTS problem for each of these output graphs separately. By Lemma 20, this last part of the algorithm can be performed in $\mathcal{O}(n + m)$ -time in total. \square

7. Conclusion

We intend the framework introduced in this paper to be applied for solving the \mathcal{H} -SQUARE ROOT problem in other graph classes — *e.g.*, graphs with *special treewidth* at most two [8]. Furthermore, we leave the existence of a full characterization of non essential cut-vertices in the square roots as an interesting open question. More generally, we aim at better understanding the relationships between small-size separators in a graph and small-diameter separators in its square. As an example, we believe that by studying the relationships between edge-separators in a graph and quasi-clique cutsets in its square (clique with one edge removed), we could improve the recognition of the squares of outerplanar graphs. Let us mention that the complexity of recognizing the squares of planar graphs is still open.

Acknowledgements

We wish to thank the referees for their careful reading of the first version of this manuscript, and their useful comments that allowed us to clarify and remove some small inaccuracies in our proofs.

References

- [1] A. Adamaszek and M. Adamaszek. Uniqueness of graph square roots of girth six. *The Electronic Journal of Combinatorics*, 18(P139):1, 2011.
- [2] G. Agnarsson and M. M. Halldórsson. Coloring powers of planar graphs. *SIAM Journal on Discrete Mathematics*, 16(4):651–662, 2003.
- [3] M. Aigner and M. Fromme. A game of cops and robbers. *Discrete Applied Mathematics*, 8(1):1–12, 1984.
- [4] A. Berry and J.-P. Bordat. Moplex elimination orderings. *Electronic Notes in Discrete Mathematics*, 8:6–9, 2001.
- [5] A. Berry, R. Pogorelcnik, and G. Simonet. Organizing the atoms of the clique separator decomposition into an atom tree. *Discrete Applied Mathematics*, 177:1–13, 2014.

- [6] A. Berry and G. Simonet. Computing the atom graph of a graph and the union join graph of a hypergraph. Technical Report arXiv:1607.02911, arXiv, 2016.
- [7] H. L. Bodlaender. A partial k -arboretum of graphs with bounded treewidth. *Theoretical computer science*, 209(1):1–45, 1998.
- [8] H.-L. Bodlaender, S. Kratsch, V. Kreuzen, O.-j. Kwon, and S. Ok. Characterizing width two for variants of treewidth. *Discrete Applied Mathematics*, 216, Part 1:29 – 46, 2017.
- [9] M. Bonamy, B. L ev eque, and A. Pinlou. 2-distance coloring of sparse graphs. *Journal of Graph Theory*, 77(3):190–218, 2014.
- [10] J. A. Bondy and U. S. R. Murty. *Graph theory*. Grad. Texts in Math., 2008.
- [11] V. Bouchitt e and I. Todinca. Treewidth and minimum fill-in: Grouping the minimal separators. *SIAM Journal on Computing*, 31(1):212–232, 2001.
- [12] M.-S. Chang, M.-T. Ko, and H.-I. Lu. Linear-time algorithms for tree root problems. In *SWAT*, pages 411–422. Springer, 2006.
- [13] S. Chaplick, M. T opfer, J. Voborn ık, and P. Zeman. On h -topological intersection graphs. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 167–179. Springer, 2017.
- [14] M. Cochefert, J.-F. Couturier, P. A. Golovach, D. Kratsch, and D. Paulusma. Parameterized algorithms for finding square roots. *Algorithmica*, 74(2):602–629, 2016.
- [15] D. Coudert and G. Ducoffe. Revisiting decomposition by clique separators. *SIAM Journal on Discrete Mathematics*, 32(1):682–694, 2018.
- [16] B. Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Information and computation*, 85(1):12–75, 1990.
- [17] P. Duchet, M. Las Vergnas, and H. Meyniel. Connected cutsets of a graph and triangle bases of the cycle space. *Discrete mathematics*, 62(2):145–154, 1986.
- [18] G. Ducoffe. Finding cut-vertices in the square roots of a graph. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 234–248. Springer, 2017.
- [19] B. Farzad and M. Karimi. Square-root finding problem in graphs, a complete dichotomy theorem. Technical Report arXiv:1210.7684, arXiv, 2012.
- [20] B. Farzad, L. C. Lau, and N. N. Tuy. Complexity of finding graph roots with girth conditions. *Algorithmica*, 62(1-2):38–53, 2012.
- [21] H. Fleischner. The square of every two-connected graph is hamiltonian. *Journal of Combinatorial Theory, Series B*, 16(1):29–34, 1974.
- [22] T. Gallai. Graphen mit triangulierbaren ungeraden vielecken. *Magyar Tud. Akad. Mat. Kutat o Int. K ozl.*, 7:3–36, 1962.

- [23] P. Golovach, D. Kratsch, D. Paulusma, and A. Stewart. A linear kernel for finding square roots of almost planar graphs. *Theoretical Computer Science*, 689:36–47, 2017.
- [24] P. Golovach, D. Kratsch, D. Paulusma, and A. Stewart. Finding cactus roots in polynomial time. *Theory of Computing Systems*, 62(6):1409–1426, 2018.
- [25] M. C. Golumbic. *Algorithmic graph theory and perfect graphs*, volume 57. Elsevier, 2004.
- [26] M. C. Golumbic and P. L. Hammer. Stability in circular arc graphs. *Journal of Algorithms*, 9(3):314–320, 1988.
- [27] F. Harary, R. M. Karp, and W. T. Tutte. A criterion for planarity of the square of a graph. *Journal of Combinatorial Theory*, 2(4):395–405, 1967.
- [28] P. Heggernes. Minimal triangulations of graphs: A survey. *Discrete Mathematics*, 306(3):297–317, 2006.
- [29] J. Hopcroft and R. Tarjan. Algorithm 447: efficient algorithms for graph manipulation. *Communications of the ACM*, 16(6):372–378, 1973.
- [30] L. C. Lau. Bipartite roots of graphs. *ACM Transactions on Algorithms (TALG)*, 2(2):178–208, 2006.
- [31] L. C. Lau and D. G. Corneil. Recognizing powers of proper interval, split, and chordal graphs. *SIAM Journal on Discrete Mathematics*, 18(1):83–102, 2004.
- [32] V. B. Le and N. T. Nguyen. A good characterization of squares of strongly chordal split graphs. *Information Processing Letters*, 111(3):120 – 123, 2011.
- [33] V. B. Le and N. N. Tuy. The square of a block graph. *Discrete Mathematics*, 310(4):734 – 741, 2010.
- [34] H.-G. Leimer. Optimal decomposition by clique separators. *Discrete mathematics*, 113(1-3):99–123, 1993.
- [35] K.-W. Lih, W.-F. Wang, and X. Zhu. Coloring the square of a K_4 -minor free graph. *Discrete Mathematics*, 269(1):303–309, 2003.
- [36] M. C. Lin, D. Rautenbach, F. J. Soullignac, and J. L. Szwarcfiter. Powers of cycles, powers of paths, and distance graphs. *Discrete Applied Mathematics*, 159(7):621–627, 2011.
- [37] Y.-L. Lin and S. S. Skiena. Algorithms for square roots of graphs. *SIAM Journal on Discrete Mathematics*, 8(1):99–118, 1995.
- [38] E. Lloyd and S. Ramanathan. On the complexity of distance-2 coloring. In *ICCI*, pages 71–74. IEEE, 1992.
- [39] R. M. McConnell. Linear-time recognition of circular-arc graphs. In *FOCS’01*, pages 386–394. IEEE.
- [40] M. Milanič and O. Schaudt. Computing square roots of trivially perfect and threshold graphs. *Discrete Applied Mathematics*, 161(10):1538–1545, 2013.

- [41] M. Molloy and M. R. Salavatipour. A bound on the chromatic number of the square of a planar graph. *Journal of Combinatorial Theory, Series B*, 94(2):189–213, 2005.
- [42] R. Motwani and M. Sudan. Computing roots of graphs is hard. *Discrete Applied Mathematics*, 54(1):81–88, 1994.
- [43] A. Mukhopadhyay. The square root of a graph. *Journal of Combinatorial Theory*, 2(3):290–295, 1967.
- [44] N. V. Nestoridis and D. M. Thilikos. Square roots of minor closed graph classes. *Discrete Applied Mathematics*, 168:34–39, 2014.
- [45] A. Oversberg and O. Schaudt. Polynomial time recognition of squares of ptolemaic graphs and 3-sun-free split graphs. In *WG*, pages 360–371. Springer, 2014.
- [46] A. Parra and P. Scheffler. Characterizations and algorithmic applications of chordal graph embeddings. *Discrete Applied Mathematics*, 79(1-3):171–188, 1997.
- [47] B. Randerath and L. Volkmann. A characterization of well covered block-cactus graphs. *Australas. J. Combin*, 9:307–314, 1994.
- [48] N. Robertson and P. Seymour. Graph minors. II. algorithmic aspects of tree-width. *Journal of algorithms*, 7(3):309–322, 1986.
- [49] I. C. Ross and F. Harary. The square of a tree. *The Bell System Technical Journal*, 39(3):641–647, 1960.
- [50] Y. Shibata. On the tree representation of chordal graphs. *Journal of Graph Theory*, 12(3):421–428, 1988.
- [51] R. E. Tarjan. Decomposition by clique separators. *Discrete Mathematics*, 55(2):221 – 232, 1985.
- [52] A. Tucker. Characterizing circular-arc graphs. *Bulletin of the American Mathematical Society*, 76(6):1257–1260, 1970.
- [53] G. Wegner. Graphs with given diameter and a coloring problem. Technical report, Univ. Dortmund, 1977.