Finding cut-vertices in the square roots of a graph

Guillaume Ducoffe

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

Guillaume Ducoffe[1,2]

[1] Université Côte d'Azur, Inria, CNRS, I3S, France
[2] National Institute for Research and Development in Informatics, 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.

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 [46]. Unsurprisingly, there is an important literature on the topic, with nice structural properties of square graphs being undercovered [2,6,15,30,33,35]. In particular, an elegant characterization of the squares of graphs has been given in [37]. 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

⋆ This work is partially supported by ANR project Stint under reference ANR-13-BS02-0007 and ANR program "Investments for the Future" under reference ANR-11-LABX-0031-01.

algorithms. They are, in fact, unlikely to exist since the problem has been proved NP-complete [36]. In light of this negative result, there has been a growing literature trying to identify the cases where the recognition of the squares of graphs remains tractable [10,22,26,25,27,32,38]. 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 [1,13,14]. 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 [26]. However, it can be done in polynomial time for many subclasses [26,27,28,34,39,43].

The most relevant examples to explain our approach are the classes of trees [43], block graphs [28] and cacti [19]. 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 [9,28,32,43], 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 [14,36]. 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. 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.

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 [19] 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 [19,28]. 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) [19]. 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 [42] 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" [18]. 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 [4].

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[3]. 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.

In Section 6, we finally answer positively to an open question of [19] 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

[3] Sufficient conditions for the framework to be applied are rather technical. They will be properly stated in a journal version.

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 [16]. 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 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 [45].

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 [9] for related work on p-th tree roots). This is left for future work. Due to lack of space, most proofs are only sketched or postponed to our technical report [11]. 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 [7]. 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* 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.

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 $dist_G(u, v)$, 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.

In what follows, we will 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 $\mathcal{H}$?

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 $dist_H(x, y) = dist_G(x, y)$ for every $x, y \in V(H)$. Furthermore, let H be a square root of a given graph $G = (V, E)$. Given a walk $\mathcal{W} = (x_0, x_1, \ldots, x_l)$ in G, an *H-extension* of $\mathcal{W}$ is any walk $\mathcal{W}'$ of H that is obtained 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 components. 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 [8,20,23,40]). 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 [19], we name v an *essential* cut-vertex if there are at least two components C_1, C_2 of $G \setminus v$ such that $C_1 \nsubseteq N_G(v)$ and similarly $C_2 \nsubseteq N_G(v)$; otherwise, v is called a *non essential* cut-vertex[4]. 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

[4] The authors in [19] 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 $C_3 \nsubseteq N_G(v)$. We do not use this notion in our paper.

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 $v \in B$. The block-cut tree of a given connected graph G can be computed in linear time [24].

It has been observed that every graph with a square root is biconnected [15]. We often use this fact in what follows.

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 [29,44]. 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. If the atoms of G are given, then the clique-atoms and the simplicial vertices of G can be computed in linear time [12]. Finally, it has been proved in [4] that the atoms of G are the bags of a tree decomposition of G, sometimes called an *atom tree*. An atom tree can be computed in $\mathcal{O}(nm)$-time, and it is not necessarily unique [4].

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 1). 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 1. *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 $u \in A \backslash A_v$. This vertex u is contained in some connected component C_u of $H \backslash v$. Furthermore since v is a cut-vertex of H, there exists $w \in N_H(v) \backslash C_u$. We claim that $S = (C_u \cap N_H(v)) \cup \{v\}$ is an uw-clique separator of G. Indeed, let us consider any uw-path $\mathcal{P}$ in G. We name $\mathcal{Q} = (x_0 = u, x_1, \ldots, x_l = w)$ 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 1 unifies and generalizes some previous results that have been found only for specific graph classes [19,28]. For example, it has been proved in [28] 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 1 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. However, there are strong ties between the two.

Theorem 1. *Let H be a square root of a given graph $G = (V, E)$. Then, the atoms of G are exactly:*

- *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 $H[B]^2$ that are not dominated in H by a cut-vertex.*

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. This notion is related to, but different than, the notion of minimal square root studied in [19][5]. The following section is based on Proposition 1, 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 1 that there is a mapping from the cut-vertices of $H_{\max}$ to the clique-atoms of its square $G = H_{\max}^2$. This mapping is injective but in general it is not surjective. 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 2. *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.

$\mathcal{C} \leftarrow \emptyset$.

for all clique-atom $A \in \mathcal{A}$ **do**

  Compute the incidence graph $I_A = Inc(\Omega(A), A)$, with $\Omega(A)$ being the multiset of neighbourhoods of the connected components of $G \setminus A$.

  if $\bigcap_{S \in \Omega(A)} S = \{v\}$ **and** v is a cut-vertex of I_A **then**

    $\mathcal{C} \leftarrow \mathcal{C} \cup \{v\};\ A_v \leftarrow A$.

The proof of Theorem 2 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 $I_A = Inc(\Omega(A), A)$ 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 1. *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 $C_1, C_2, \ldots, C_k$ be all the connected components of $H \setminus v$ such that $C_i \not\subseteq A_v$. Note that $k \geq 2$ by the hypothesis. 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 $dist_H(u_i, u_j) = dist_H(u_i, v) + dist_H(u_j, v) \geq 4$, 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, let us group the components of $G \setminus A_v$ in order to obtain the sets $C_i \setminus A_v,\ 1 \leq i \leq k$. 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 $N_G(P) \cap N_G(Q) = \{v\}$. $\square$

[5] Let $\mathcal{H}$ be 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.

It turns out that conversely, Lemma 1 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 2. *Let $H_{\max}$ be a maxblock square root of a given graph $G = (V, E)$, and let $v \in V$. Suppose 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 $N_{H_{\max}}[v] = A_v$.*

Correctness of Algorithm 1 follows from Lemmas 1 and 2. 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.

4.2 Sufficient conditions for non essential cut-vertices

We let 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. 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 3).

Non essential cut-vertices 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 such a cut-vertex. 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 3. *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, $N_{H_{\max}}[v] = N_G[u]$.*

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

Theorem 3. *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: $H_{\max}[B]$ has no dominated vertex, unless B is a clique*[6]*; and $H_{\max}[B]^2$ 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.*

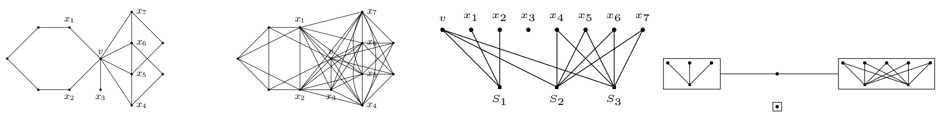
[6] This first assumption on the blocks may look a bit artificial. However, we emphasize that it holds for every regular graph [3].

Sketch proof. Let H be any square root of G with its blocks satisfying the two assumptions of the theorem. By analogy between the block-cut tree of H and an atom tree of G, it can be shown that the closed neighbourhood of a non essential cut-vertex in H satisfies the condition of the theorem. Conversely, if a clique-atom of G is a leaf in some atom tree, then either it is the closed neighbourhood of some (non essential) cut-vertex, or it is the square of a block B of H with diameter two. In the latter case, we deduce from the hypothesis – that there can be no dominated vertex in B – that B must contain a single cut-vertex v of H. Let us pairwise connect all the neighbours of v in B. Then, let us make of all the remaining vertices in $B \setminus N_H[v]$ a set of pending vertices adjacent to an arbitrary neighbour $u \in N_H(v) \cap B$. In doing so, we keep the property to be a square root of G and we strictly increase the number of blocks. $\square$

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 how to compute for every block of $H_{\max}$ a graph that is isomorphic to its square.

Theorem 4. *Let $H_{\max}$ be a maxblock square root of a graph $G = (V, E)$, and let $A_1, A_2, \ldots, A_k$ 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 can also compute the mapping from $V(G_B)$ to B. It can be done in $\mathcal{O}(n+m)$-time in total if an atom tree of G is given.*



(a) Square root H. (b) Square $G = H^2$. (c) Incidence graph. (d) Block-cut tree.

Fig. 1: Computation of the connected components in a square root.

Sketch proof. This part reuses the same techniques as Section 4.1. 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 two 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 1 for an illustration of the procedure. Furthermore, it can be proved that when A is the closed-neighbourhood of a cut-vertex v in $H_{\max}$, the equivalence classes of $\equiv_A$ are exactly the sets $C_i \setminus A,\ 1 \leq i \leq l$, with $C_1, \ldots, C_l$ being the connected components of $H_{\max}$. Applying this procedure sequentially to all the clique-atoms that represent the closed neighbourhood of a cut-vertex in $H_{\max}$, we can compute the squares of each block of $H_{\max} \setminus v$. This can be done in total $\mathcal{O}(n+m)$-time by carefully using the adhesion sets in an atom tree of G. □

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 sequence $v_1, v_2, \ldots, v_k \in V$ of pairwise distinct 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 $v_1, \ldots, v_k$ to be computed. Notice that we need to ensure that all the v_i's are distinct 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.

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 [41]; and the *Gallai trees*, that are the cactus-block graphs with no block being isomorphic to an even cycle [16]. Our main result in this section is that the squares of these graphs can be recognized in polynomial time:

Theorem 5. *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 5 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 6. *Squares of cactus-block graphs, resp. squares of Gallai trees, can be recognized in $\mathcal{O}(nm)$-time.*

(a) A pending block. (b) Non-essential cut. (c) Splittable block. (d) Essential cut.

Fig. 2: Local modifications of the blocks.

The proof of Theorem 5 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 show that the cut-vertices in this square root are exactly those characterized by Theorems 2 and 3. We do so by adapting the respective techniques from Lemma 2 and Theorem 3 in order to increase the number of cut-vertices. An illustration is provided with Figure 2. Then, we need to show that $\mathcal{H}$-SQUARE ROOT WITH NEIGHBOURS CONSTRAINTS can be solved in linear time for j-cactus-block graphs. 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 [21,31].

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 [5]. 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 [17]. Let us mention that the complexity of recognizing the squares of planar graphs is still open.

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, R. Pogorelcnik, and G. Simonet. Organizing the atoms of the clique separator decomposition into an atom tree. *Discrete Applied Mathematics*, 177:1–13, 2014.
5. 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.
6. M. Bonamy, B. Lévêque, and A. Pinlou. 2-distance coloring of sparse graphs. *Journal of Graph Theory*, 77(3):190–218, 2014.
7. J. A. Bondy and U. S. R. Murty. *Graph theory.* Grad. Texts in Math., 2008.
8. V. Bouchitté and I. Todinca. Treewidth and minimum fill-in: Grouping the minimal separators. *SIAM Journal on Computing*, 31(1):212–232, 2001.
9. M.-S. Chang, M.-T. Ko, and H.-I. Lu. Linear-time algorithms for tree root problems. In *SWAT*, pages 411–422. Springer, 2006.
10. 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.
11. G. Ducoffe. Finding cut-vertices in the square roots of a graph. Technical Report hal-01477981, UCA, Inria, CNRS, I3S, France, 2017. Available at `https://hal.archives-ouvertes.fr/hal-01477981`.
12. G. Ducoffe and D. Coudert. Clique-decomposition revisited. In revision (Research Report on HAL, hal-01266147), 2017.
13. B. Farzad and M. Karimi. Square-root finding problem in graphs, a complete dichotomy theorem. Technical Report arXiv:1210.7684, arXiv, 2012.
14. B. Farzad, L. C. Lau, and N. N. Tuy. Complexity of finding graph roots with girth conditions. *Algorithmica*, 62(1-2):38–53, 2012.
15. H. Fleischner. The square of every two-connected graph is hamiltonian. *Journal of Combinatorial Theory, Series B*, 16(1):29–34, 1974.
16. T. Gallai. Graphen mit triangulierbaren ungeraden vielecken. *Magyar Tud. Akad. Mat. Kutató Int. Közl*, 7:3–36, 1962.
17. P. Golovach, P. Heggernes, D. Kratsch, P. Lima, and D. Paulusma. Algorithms for outerplanar graph roots and graph roots of pathwidth at most 2. In *WG*. Springer, 2017. available on arXiv:1703.05102.
18. P. Golovach, D. Kratsch, D. Paulusma, and A. Stewart. A linear kernel for finding square roots of almost planar graphs. In *SWAT*, pages 4:1–4:14, 2016.
19. P. A. Golovach, D. Kratsch, D. Paulusma, and A. Stewart. Finding cactus roots in polynomial time. In *IWOCA*, pages 361–372. Springer, 2016.
20. M. C. Golumbic. *Algorithmic graph theory and perfect graphs*, volume 57. Elsevier, 2004.
21. M. C. Golumbic and P. L. Hammer. Stability in circular arc graphs. *Journal of Algorithms*, 9(3):314–320, 1988.
22. 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.

23. P. Heggernes. Minimal triangulations of graphs: A survey. *Discrete Mathematics*, 306(3):297–317, 2006.
24. J. Hopcroft and R. Tarjan. Algorithm 447: efficient algorithms for graph manipulation. *Communications of the ACM*, 16(6):372–378, 1973.
25. L. C. Lau. Bipartite roots of graphs. *ACM Transactions on Algorithms (TALG)*, 2(2):178–208, 2006.
26. 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.
27. 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.
28. V. B. Le and N. N. Tuy. The square of a block graph. *Discrete Mathematics*, 310(4):734 – 741, 2010.
29. H.-G. Leimer. Optimal decomposition by clique separators. *Discrete mathematics*, 113(1-3):99–123, 1993.
30. 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.
31. M. C. Lin, D. Rautenbach, F. J. Soulignac, and J. L. Szwarcfiter. Powers of cycles, powers of paths, and distance graphs. *Discrete Applied Mathematics*, 159(7):621–627, 2011.
32. Y.-L. Lin and S. S. Skiena. Algorithms for square roots of graphs. *SIAM Journal on Discrete Mathematics*, 8(1):99–118, 1995.
33. E. Lloyd and S. Ramanathan. On the complexity of distance-2 coloring. In *ICCI*, pages 71–74. IEEE, 1992.
34. M. Milanič and O. Schaudt. Computing square roots of trivially perfect and threshold graphs. *Discrete Applied Mathematics*, 161(10):1538–1545, 2013.
35. 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.
36. R. Motwani and M. Sudan. Computing roots of graphs is hard. *Discrete Applied Mathematics*, 54(1):81–88, 1994.
37. A. Mukhopadhyay. The square root of a graph. *Journal of Combinatorial Theory*, 2(3):290–295, 1967.
38. N. V. Nestoridis and D. M. Thilikos. Square roots of minor closed graph classes. *Discrete Applied Mathematics*, 168:34–39, 2014.
39. 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.
40. A. Parra and P. Scheffler. Characterizations and algorithmic applications of chordal graph embeddings. *Discrete Applied Mathematics*, 79(1-3):171–188, 1997.
41. B. Randerath and L. Volkmann. A characterization of well covered block-cactus graphs. *Australas. J. Combin*, 9:307–314, 1994.
42. N. Robertson and P. Seymour. Graph minors. II. algorithmic aspects of tree-width. *Journal of algorithms*, 7(3):309–322, 1986.
43. I. C. Ross and F. Harary. The square of a tree. *The Bell System Technical Journal*, 39(3):641–647, 1960.
44. R. E. Tarjan. Decomposition by clique separators. *Discrete Mathematics*, 55(2):221 – 232, 1985.
45. A. Tucker. Characterizing circular-arc graphs. *Bulletin of the American Mathematical Society*, 76(6):1257–1260, 1970.
46. G. Wegner. Graphs with given diameter and a coloring problem. Technical report, Univ. Dortmund, 1977.