Algorithms for Square Roots of Graphs

Yaw-Ling Lin*
Steven S. Skiena†

Department of Computer Science
State University of New York
Stony Brook, NY 11794-4400

Abstract

The *n-th power* ($n \geq 1$) of a graph $G = (V, E)$, written G^n, is defined to be the graph having V as its vertex set with two vertices u, v adjacent in G^n if and only if there exists a path of length at most n between them. Similarly, graph H has an *n-th root* G if $G^n = H$. For the case of $n = 2$, we say that G^2 is the *square* of G and G is the *square root* of G^2. Here we give a linear time algorithm for finding the tree square roots of a given graph and a linear time algorithm for finding the square roots of planar graphs. We also give a polynomial time algorithm for finding the square roots of subdivision graphs, which is equivalent to the problem of the inversion of total graphs. Further, we give a linear time algorithm for finding a Hamiltonian cycle in a cubic graph, and we prove the NP-completeness of finding the maximum cliques in powers of graphs and the chordality of powers of trees.

Keywords: Square graphs, power graphs, tree square, planar square graphs.

1 Introduction

Given an undirected graph G, the *distance* between vertices u and v, denoted by $d_G(u, v)$, is the length of the shortest path from u to v in G. The *n-th power* ($n \geq 1$) of G, written G^n, is defined to be the graph having $V(G)$ as its vertex set with two vertices u, v adjacent in G^n if and only if there exists a path of length at most n between them, i.e., $d(u, v) \leq n$. Similarly, graph H has an *n-th root* G if $G^n = H$. For the case of $n = 2$, we say that G^2 is the *square* of G and G is the *square root* of G^2.

Mukhopadhyay [21] found that a connected undirected graph G with vertices $v_1, ..., v_n$ has a square root if and only if G contains a collection of n complete subgraphs $G_1, ..., G_n$ such that for all $1 \leq i, j \leq n$:

*The first author's current address is: Department of Information Science, Providence University, 200 Chung Chi Road, Sa-Lu, Taichung Shang, Taiwan. E-mail: yllin@host1.pu.edu.tw

†The work of the second author was partially supported by NSF Research Initiation Award CCR-9109289. E-mail: skiena@sbcs.sunysb.edu

1. $\bigcup_{1\leq i\leq n} G_i = G$,

2. $v_i \in G_i$,

3. $v_i \in G_j$ if and only if $v_j \in G_i$.

Characterizations of squares of digraphs were given by Geller [13] and of n-th power of graphs and digraphs was given by Escalante, Montejano, and Rojano [7]. Ross and Harary [23] showed that a graph has an unique tree square root (up to isomorphism) if it has a tree as its square root.

Although many properties of the square roots of graphs have been discovered, no efficient algorithms for finding the square root of a graph was known. One reason is that the mathematical characterization of the squares of graphs involve the concept of *clique* which is an NP-complete problem in general. Furthermore, note that the characterization given by Mukhopadhyay dealing with *some* n cliques of G which will have exponentially many candidate cliques. Indeed, Motwani and Sudan [20] showed that recognizing the general square graphs is an NP-complete problem.

Fleischner [8] proved that the square of a biconnected graph is always Hamiltonian. Although the Hamiltonian cycle problem is NP-complete for general graphs [10], the biconnectivity of a graph can be tested in linear time [25]. Thus an efficient algorithm for finding the square root of some special classes of graphs could be useful for finding Hamiltonian cycles in these graphs. Unfortunately, Underground [27] had shown that finding a Hamiltonian cycle in G^2 given G is also NP-complete.

In the first part of this paper, we concentrate on the problem of finding the square roots of graphs. We give an $O(m)$ time algorithm for finding the tree square roots of a graph in Section 2, and a linear time algorithm for finding the square roots of a planar graph in Section 3. In Section 4, we propose an $O(m^2)$ time algorithm for inversion of total graphs, which are the square of subdivided graphs.

Optimization problems concerning the square, and more generally the powers, of graphs are discussed in Section 5. We prove the following results:

- Finding the maximum clique, the minimum independent set, the minimum dominating set, the minimum clique partition, and the chromatic number in G^k are all NP-complete for any fixed $k \geq 1$. Also, we show that when G is a tree, G^k is a chordal graph for any $k \geq 1$, implying these five optimization problems in G^k can all be determined in linear time [14].

- A Hamiltonian cycle in G^k, $k \geq 3$, can be found in linear time given G. When G is a tree, the existence of a Hamiltonian cycle in G^2 can be determined in linear time.

We conclude with open problems.

2 Tree Square Roots

Tree square roots were first considered by Ross and Harary [23], who showed that they are unique up to isomorphism. Figure 1 presents a tree and its square. In this section,

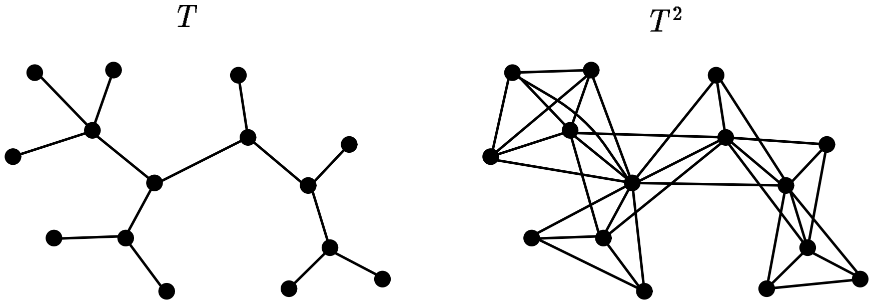

Figure 1: A tree and its square.

we present an $O(|V| + |E|)$ algorithm for finding the tree square root of a given graph $G = (V, E)$. We will show that the leaves of the tree square root can be readily identified, and, after all leaves of its square root have been found, we can trim all leaves from the graph so resulting in a graph that is the square of a smaller tree. Repeating this process and building appropriate edges between the different levels of leaves, the ultimate structure of its tree square root follows.

Now consider a graph $T^2 = (V, E)$ which is known to be the square of some tree $T = (V, E')$. The *degree* of a vertex v in G, written $deg_G(v)$, is the size of its neighborhood. A vertex v is a *leaf* (or *endpoint*) of a graph G if $deg_G(v) = 1$.

The longest distance among all vertices of a graph is called its *diameter*. A tree with diameter less than or equal to 2 is called a *star*. Note that the square of a star is a complete graph. Recall that a vertex v is simplicial if its neighborhood forms a clique.

Lemma 2.1 *v is a simplicial point of T^2 if and only if v is a leaf of T or T is a star.*

Proof. Assume that v is a leaf of T. Since $N'_{T^2}(v) = \{u : d_T(u, v) \leq 2\}$, graph $\langle N'_{T^2}(v)\rangle_T$ can be obtained by starting from a leaf V and walking at most two steps in T. The resulting graph has a diameter less than or equal to two , i.e., $\langle N'_{T^2}(v)\rangle_T$ is a star. The square of a star is a complete graph, so $\langle N'_{T^2}(v)\rangle_{T^2}$ is complete, i.e., v is simplicial. Now assume v is not a leaf and T is not a star. Then there must be a path (x, v, y, z) in T. Since $d_T(x, z) = 3$ implies $xz \notin E(T^2)$ and $x, z \in N_{T^2}(v)$, v is not simplicial. ▮

For each leaf v of T, we can partition $N_{T^2}(v)$ into two sets L_v and M_v, where L_v denotes all leaves of T in $N_{T^2}(v)$ and M_v the set $N_{T^2}(v) - L_v$. According to Lemma 2.1, a leaf node has the lowest degree of any vertex in its neighborhood of T^2. The following lemma characterizes when the degree of a leaf will be strictly less than the degree of its neighboring internal node.

Lemma 2.2 *Let T be a non-star tree and v a leaf of T, then*

1. *For all $u \in L_v, deg_{T^2}(u) = deg_{T^2}(v)$.*
2. *For all $w \in M_v, deg_{T^2}(w) > deg_{T^2}(v)$.*

Proof. For part 1, let $u \in L_v$. Since $\langle N'_{T^2}(v)\rangle_T$ is a star, we know that $N'_{T^2}(v) = N'_{T^2}(u)$. So $deg_{T^2}(u) = deg_{T^2}(v)$. For part 2, let $w \in M_v$. Since T is not a star, without loss of generality, there is either a path (v, w, x, y) or a path (v, x, w, y) in T. For either case, $y \in N_{T^2}(w)$ but $y \notin N_{T^2}(v)$. We conclude that $N'_{T^2}(w) \not\subset N'_{T^2}(v)$. Since, by Lemma 2.1, $N'_{T^2}(v) \subset N'_{T^2}(w)$, $deg_{T^2}(w) > deg_{T^2}(v)$. ∎

The *center* of star T is the central vertex v such that for all other vertices u in T, $d_T(u, v) = 1$. If $|T| \neq 2$, since there is only one center v in T, we can denote v by $center(T)$. Let K_n denote the complete graph of size n. The size of M_v, $|M_v|$, provides some valuable information.

Lemma 2.3 *Let v be a leaf of T, then*

1. *if $|M_v| \leq 1$, T is a star;*
2. *if $|M_v| = 2$, say $M_v = \{x, y\}$, then $xy \in E(T)$;*
3. *if $|M_v| \geq 3$, $(v, center(M_v)) \in E(T)$.*

Proof. For part 1, either $|M_v| = 0$ meaning $T = K_2$ or $|M_v| = 1$, where again, T is a star. For part 2, let $M_v = \{x, y\}$. Since v is a leaf, Lemma 2.1 showed that $\langle N'_{T^2}(v)\rangle_T$ is a star. Since $|N_{T^2}(v)| \geq |M_v| = 2$ meaning $|N'_{T^2}(v)| \geq 3$, $center(\langle N'_{T^2}(v)\rangle_T)$ is well-defined. Further, $center(\langle N'_{T^2}(v)\rangle_T) \notin L_v$. So, without loss of generality, we can let x be the center of $\langle N'_{T^2}(v)\rangle_T$. Again, since $\langle N'_{T^2}(v)\rangle_T$ is a star, y must be connected to x. So $xy \in E(T)$. For part 3, $|M_v| \geq 3$. Since $\langle N_{T^2}(v)\rangle_T$ is a star and v is the center, we know that $center(\langle N_{T^2}(v)\rangle_T) = center(\langle M_v\rangle_T)$. Clearly $(v, center(\langle N_{T^2}(v)\rangle_T)) \in E(T)$. ∎

Given a tree T, we can delete all the leaves of T resulting in a smaller tree T'. This trimming operation defines a function $trim(T) = T'$. The trimming operation can be repeated until the remaining subgraph T' is empty. Since each vertex v in T will eventually be trimmed, we can associate an integer with v specifying the number of trimming operation taken before v becomes a leaf. This function from $V(T)$ to $[0..\lfloor(n-1)/2\rfloor]$ is recursively defined as follow:

$$level(v) = \begin{cases} 0 & \text{if } v \text{ is a leaf of } T \\ k & \text{if } v \text{ is a leaf of } trim^k(T) \end{cases}$$

Lemma 2.1 shows that a leaf has the lowest degree in its neighborhood. The following lemma shows the opposite direction. A vertex having the lowest degree in its neighborhood is not always a leaf but we can easily distinguish a low degree internal node from a leaf:

Lemma 2.4 *Let T be a non-star tree and v a non-leaf vertex of T. If $deg_{T^2}(v) \leq min\{deg_{T^2}(u) : u \in N_{T^2}(v)\}$, then $level(v) > 2$ and for all $x \in N_{T^2}(v), N_{T^2}(v) \not\subset N'_{T^2}(x)$.*

Proof. Since v is not a leaf, $level(v) \geq 1$. Suppose $level(v) \leq 2$, then there must exist a u, a leaf of T, such that $uv \in E(T^2)$. By Lemma 2.2, we know that $deg_{T^2}(u) < deg_{T^2}(v)$, a contradiction. Let $x \in N_{T^2}(v)$. The second part has two cases:

Case 1: $d_T(x, v) = 1$. Since T is not a star and $level(v) > 2$, there exists a path (x, v, y, z) in T. Now, $z \in N_{T^2}(v)$ but $z \notin N'_{T^2}(x)$.

Case 2: $d_T(x,v) = 2$. Since v is not a leaf, there must be a path (x,y,v,z) in T. Again, $z \in N_{T^2}(v)$ but $z \notin N'_{T^2}(x)$. ∎

Lemma 2.5 *Given a tree square $T^2 = (V,E)$, we can identify all leaves of T by examining T^2 in a total time of $O(|E|)$ time.*

Proof. Assume that the graph T^2 is given in form of adjacency lists. Let $|V| = n$, and $|E| = m$. First we can calculate the degree of each vertex totally in $O(m)$ time. Also, we will maintain a boolean array B, indexed by the vertices V, with size n, and initialize it by all zeros in $O(n)$ time. Then, for each vertex v of T^2, first we will check whether v has the lowest degree among its neighborhoods in $O(deg_{T^2}(v))$ time. If not, v cannot be a leaf in T; otherwise, we can mark $B_u = 1$ for each u in $N_{T^2}(v)$ within $O(deg_{T^2}(v))$ time, and choosing u such that u has the lowest degree in $N_{T^2}(v)$, check whether $N_{T^2}(v) \subset N_{T^2}(u)$ in $O(deg_{T^2}(u))$ time, by polling each elements of $N_{T^2}(u)$ in B_v. If this condition does not hold, by Lemma 2.5, no vertex of $N'_{T^2}(v)$ is a leaf; otherwise, v and those vertices in $N_{T^2}(v)$ with same degree as v are all leaves of T. Note that the total time spent on admitting leaves is bounded by $\Sigma_{level(v)\in\{0,1\}} deg_{T^2}(v)$. Also note that, since we are picking u, the lowest degree vertex in $N_{T^2}(v)$, the time spent on asking $N_{T^2}(v) \subset N_{T^2}(u)$ for those failed (nonleaf) v is bounded by $deg_{T^2}(x)$ for any vertex $x \in N_T(v)$. Further, if v_1 and v_1 both failed this test, it implies that $d_T(v_1,v_2) > 2$, and, for each leaf y of T, both $d(y,v_1)$ and $d_T(y,v_2)$ are greater than 2. That is, the amount of time failing the $N_{T^2}(v) \subset N_{T^2}(u)$ test is bounded by $\Sigma_{uv\in F} deg_{T^2}(u) + deg_{T^2}(v)$ for some $F \subset E(T)$ such that for each two edges of F, they are not adjacent to each other. It follows that the total time is bounded by $\Sigma_{v\in T^2} deg_{T^2}(v)$, or $O(|E|)$. ∎

Theorem 2.6 *The tree square root of a graph can be found in $O(m)$ time, where m denotes the number of edges of the given tree square graph.*

Proof. Let the given graph be $G = (V,E)$ with $|V| = n$ and $|E| = m$. Now assume that there is a tree T such that $G = T^2$. By Lemma 2.5, we can identify all leaves of T in $O(m)$ time. Then we repeat the following procedures:

Let $LF(T)$ denote the set of all leaves of tree T. Trimming T^2 by removing all vertices in $LF(T)$ results in a proper induced subgraph of T^2, say T'^2. This process takes $O(\sum_{v\in LF(T)} deg_G(v))$ time. For all leaf v, we can partition its neighborhood into two sets L_v and M_v totally in $O(\sum_{v\in LF(T)} deg_G(v))$ time. If for all v, $M_v < 2$, we will conclude that T is a star and terminate the process. Else, if for all v, $M_v > 2$, then G is not a square of a tree; otherwise, we have some $v \in LF(T)$ such that $M_v = \{x,y\}$, a set of size two. According to the second part of Lemma 2.3, we can report that edge xy contains in T. Further, we claim that

$$LF(T') \subset \bigcup_{v\in LF(T)\wedge |M_v|=2} M_v.$$

To prove this claim, assume that p is a leaf of T', meaning there exists exactly one vertex, say q, in T' such that $pq \in T'$. Since p is a leaf, there must be a leaf v in T adjacent to

p; otherwise, p will be a leaf of T meaning $p \notin T'$, a contradiction. That is, $\{p, q\} = M_v$. Our claim is proven.

Our claim means that we can find all the leaves of T' in $O(|LF(T)|)$ time by Lemma 2.4, since we will now only need to check which vertex of $\{p, q\}$ has lower degree. Treat T' as T and repeat the process above until T'^2 becomes a complete graph, implying T' a star. Each pass takes $O(\sum_{v \in LF(T)} deg_G(v))$ to identify the leaves of T and establish the set of possible leaves of T'. If G is a tree square, eventually each vertex v of G becomes a leaf during the trimming process. That is, totally we will spend $O(\sum_{v \in V(G)} deg_G(v))$ or $O(m)$ time for finding all *internal* edges (those edges which are not adjacent with leaves) of T.

At this point, only the original leaves have not yet been linked to other vertices. This problem can be solved by recording the internal nodes, M_v, of each original leaf v of T. Along with the trimming process, the trimmed level of each vertex v of T, $level(v)$, is recorded. In detail, we maintain a counter, say k, initialized by zero. Every time the trimming process is repeated, k is increased by one. Each time when a vertex v is eliminated from the tree, we will record $level(v)$ as k. Note that this fulfills our definition of $level$ function and takes totally $O(n)$ time.

Recall that the structure of the internal tree has already been determinated at this point. If G is complete, T is just a star, which case we have already dealt with when we try to find the leaves. Now we may assume that G is not complete, meaning $|M_v| \geq 2$. More precisely, the size of M_v can be:

Case 1: $|M_v| = 2$ or $M_v = \{x, y\}$. If $level(x) = level(y) = 1$, there will be exactly two set of leaves, say L_1 and L_2, and we can either link x to L_1 and y to L_2 or y to L_1 and x to L_2. Otherwise, exactly one of x and y is in the level 1. Without loss of generality, we say $level(x) = 1$, then v and x are linked. This case takes constant time.

Case 2: $|M_v| \geq 3$. By Lemma 2.3, we shall link v to $center(M_v)$. Now we show how the $center(M_v)$ is found: First choose two arbitrary x, y from M_v. If $xy \in T'$, pick another z from M_v and since $\langle M_v \rangle_{T'}$ is a star, so z is adjacent to, say, x. That is, x is the center. If $xy \notin T'$. Keep picking z from M_v and eventually, since $\langle M_v \rangle_{T'}$ is a star, we will find a z such that zx and $zy \in T'$. z is the center and the search takes $|M_v|$ or $O(deg_G(v))$ time.

The process of linking the leaves of T to T' takes $O(\sum_{v \in LF(T)} deg_G(v))$, also bounded by $O(m)$. After the tree T has been found, we will now double check whether $G = T^2$. This can be done by examining each vertex v of T and see whether $\bigcup_{u \in N_T} N_T(u) \cup N_T(v) = N_G(v)$ in $O(deg_G(v))$ time, since the time needed for finding $\bigcup_{u \in N_T} N_T(u)$ is bounded by $2deg_G(v)$ by the trick of using a boolean array as we have shown in Lemma 2.5. So totally, it will take $O(m)$ time to finish the last check.

Now we have completed the analysis and description of the algorithm. In summary, it takes $O(m)$ time to identify all leaves of T, and another $O(m)$ time to trim the tree and find the structure of T. Finally, to make sure T is indeed the square root of G, it needs another $O(m)$ to do so. So we now conclude that the tree square root of a square graph can be found in linear time. ∎

3 Square Roots of Planar Graphs

In this section, we present an $O(n)$ algorithm for finding the square root of a given planar graph based on the characterization of planar squares found by Harary, Karp, and Tutte [16] and the linear time triconnected components algorithm given by Hopcroft and Tarjan [18].

A vertex v is an *articulation vertex* (or *cut point*) of a connected graph G if the removal of v from G results in a disconnected graph. A *bridge* of a connected graph G is an edge $e \in E(G)$ whose removal disconnect G. A *block* or *biconnected component* of G is a maximal connected induced subgraph of G without articulation vertices. A *n-connected* graph G is a graph with at least $n+1$ vertices such that the removal of any $n-1$ vertices does not disconnect G [26]. Note that the complete graph K_n is $(n-1)$-connected but not n-connected. A graph is *biconnected* (*triconnected*) if it is 2-connected (3-connected.)

Theorem 3.1 (Harary, Karp, and Tutte [16]) *A graph G has a planar square if and only if*

1. *every point of G has degree less than or equal to three,*
2. *every block of G with more than four points is a cycle of even length, and*
3. *G does not have three mutually adjacent articulation vertices.*

The following discussion concerns the definition of *triconnected components* given by Hopcroft and Tarjan [18]. Let $G = (V, E)$ be a biconnected graph. A pair of vertices $\{a, b\}$ in G is called a *separation pair* of G if the removal of $\{a, b\}$ disconnect G. Let $S = \{H_i : H_i$ is a connected component of $\langle V - \{a,b\}\rangle_G\}$. and $\mathcal{S} = \{G_i = \langle V(H_i) \cup \{a,b\}\rangle_G$ with ab linked: for each $H_i \in S\}$. Each element G_i of $\mathcal{S}$ is called the *split graph* of G with respect to $\{a, b\}$. The newly added edge ab, called *virtual edge*, is not necessarily in G. Given a biconnected but not triconnected or complete graph G, we can find one of its separation pairs, $\{a, b\}$ and split it into split graphs. If any split graph G_i of G is not triconnected or complete, the splitting process is repeated until each split graph is either triconnected or complete. The ultimate split graphs are called the *split components* of G. The split components are not necessarily unique. Given a set of split graphs, partition it into two classes: $\mathcal{A} = \{G_i : G_i$ a triconnected split component$\}$ and $\mathcal{T} = \{G_i : G_i$ a split component and $G_i = K_3\}$ *Merge* those split components in $\mathcal{T}$ that share edges into a polygon. Let $\mathcal{P}$ denote the collection of the resulting in components. $\mathcal{A} \cup \mathcal{P}$ is called the *triconnected components* of G. Triconnected components are unique and

Theorem 3.2 (Hopcroft and Tarjan [18]) *The triconnected components of a graph can be found in $O(m + n)$ time.*

An *endline* of G is an edge uv of G such that either u or v is a leaf. A *burr* of G is a maximal connected (induced) subgraph of G in which every bridge is an end line. By removing all leaves of a burr B in G, the remaining subgraph B' is called the *central block* of B. Since B does not contain inner bridge, B' must be a block or a single vertex. As shown in Theorem 3.1, if G^2 is a planar graph, then for each burr B of G, either

Case 1. The central block B' is an even length cycle with more than four vertices. Let C_n denote a chordless cycle of length n. That is, $B' = C_{2n}$, for some n greater than 2. Further, each vertex of B' is adjacent to at most one endline.

Case 2. The central block B' is a block of size at most four. Each vertex of B' can be linked with at most one leaf, providing it does not violate the conditions of Theorem 3.1. Such kind of burr has at most eight vertices. That is, a burr consists of a central block C_4 with each vertex adjacent to exactly one leaf.

Case 3. The central block B' consists of a single vertex v, which can be adjacent to at most three other leaves. That is, B is a star of size at most four.

Lemma 3.3 *The square of a burr is triconnected or complete; Further, each triconnected component of a planar square G^2 is the square of a burr in G.*

Proof. Given a graph $G = (V, E), U \subset V, v \in V$, and $x \notin V$, let $G - U$ denote $\langle V - U \rangle_G$ and $G \cup ax$ denote the graph $(V \cup \{x\}, E \cup ax)$. First we prove that the square of a burr is triconnected or a complete graph with at most four vertices. Given a burr B, we analyze the cases of different burrs based on the size of its central block B'.

Case 1. B' is a block of size greater or equal to three. We claim that:

(1) B'^2 is triconnected or K_3. If $|B'| = 3$, $B'^2 = K_3$. If $|B'| = 4$, $B'^2 = K_4$ is triconnected. Otherwise, $B' = C_{2n}$ for some $n > 2$. Consider each pair of vertices, $\{u, v\}$, of C_{2n}. If $uv \in E(C_{2n})$, then $C_{2n} - \{u, v\}$ is still connected. If $uv \notin E(C_n)$, without loss of generality, let $C_{2n} = (P_1, u, P_2, v)$, where P_1 and P_2 represent two different paths in C_{2n}. Note that in C_{2n}^2, there exist at least one edge that connects P_1 and P_2. That is, $C_{2n}^2 - \{u, v\}$ is connected or B'^2 is triconnected.

(2) For each end line vx of B, $(B' \cup xv)^2$ is triconnected. Let x be the leaf of B meaning $v \in B'$. For each $w \in B'$, delete $\{x, w\}$ from $(B' \cup xv)^2$ results in a graph $B'^2 - \{w\}$, which is still connected. Otherwise delete $\{u, v\} \subset V(B')$ from $(B' \cup xv)^2$. $B'^2 - \{u, v\}$ is still connected as shown in (1). Note that x is adjacent to at least three vertices in B'^2. So there must exist another vertex $w \in V(B') - \{u, v\}$ that is adjacent to x at the $(B' \cup vx)^2$.

By the claim (1) and (2) stated above, we can easily see that if $B \neq K_3$, B is triconnected. Since the only case we have not mentioned is picking the vertices pair $\{u, v\}$ such that both of them are the leaves, but clearly $B - \{u, v\}$ is still connected.

Case 2. The central block is a single vertex. Such burr must be a star of size at most four. The square of stars are complete as shown in Lemma 2.1.

Now we proceed to prove that each triconnected component of G^2 is the square of a burr. First we note that vertex $\{u, v\}$ of G^2 is a separation pair of G^2 if and only if uv is the non-endline bridge of G. So the split process will be repeated until the resulting graphs are triconnected or complete graphs with at most three vertices. Suppose there is a triconnected component H^2 in G^2, such that H is not a burr of G. By the definition of burr, there exists a non-endline bridge uv in H. Since the removal of vertices pair $\{u, v\}$ disconnects H^2, which means that the splitting process has not completed, we conclude that H^2 is not the triconnected component of G^2. ∎

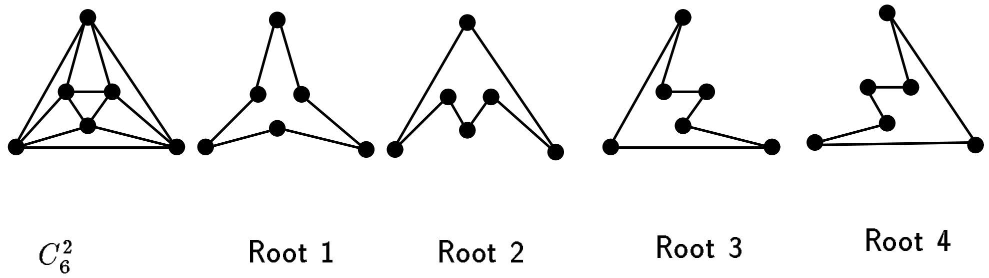

Figure 2: Four different roots of C_6^2.

Theorem 3.2 shows that $O(m+n)$ time suffices to find all triconnected components of a given graph. For our case of planar graph, because the number of edges $m \leq 3n-6$, we can find all triconnected components in $O(n)$ time.

For G^2 to be planar, each large burr in G must be an even length cycle plus some endlines in general. Otherwise, the central block will just be a finite graph with size less than five. The following lemma characterizes the leaves of a large burr.

Lemma 3.4 *Given a planar square graph G^2 and a burr B of G such that $|B| > 5$, then v is a leaf of B if and only if $deg_{B^2}(v) = 3$.*

Proof. Since $|B| > 5$ and B^2 is planar, the central block B' must have at least four vertices. Otherwise the condition of Theorem 3.1 will be violated. First we show that for each non-leaf v of B, $deg_{B^2}(v) > 3$. If B' is an even length cycle C_{2n} such that $n \geq 3$, then $deg_{B^2}(v) \geq 4$. Otherwise, the central block is a block of size four. There are at least two leaves outside this size-four block. Again, $deg_{B^2}(v) \geq 4$.

Now we show the each leaf $v \in B$ has degree three in B^2. Let $uv \in B$ meaning $u \in B'$. Since B^2 is planar, $deg_B(u) \leq 3$. Since B' is a block, $deg_{B'}(u) \geq 2$, meaning $deg_B(u) \geq 3$. Since $deg_B(u) = 3$, $deg_{B^2}(v) = 3$. ▮

Now we show that we can efficiently find the original burr by given its square.

Lemma 3.5 *Given a planar graph B^2, where B is a burr, the structure of B can be computed in $O(|B|)$ time.*

Proof. Consider various sizes of $|B|$.

Case 1. $|B| > 5$. By Lemma 3.4, we can identify all leaves of B within $O(|B|)$ time by examining the degree of each vertex. Let L be the set of leaves in B. There can be two possibilities:

Case 1.1. $|L| = 0$. Then $B = C_{2n}$ for some $n \geq 3$. If $B = C_6$, there are 4 distinct labeled graphs whose square is equal to C_6^2 as shown in Figure 2. For $n > 3$, the identification of C_{2n} are unique since edge $uv \in C_{2n}$ if and only if $|N'_{C_{2n}^2}(u) \cap N'_{C_{2n}^2}(v)| = 3$. In linear time we can trace the exact labeling of C_{2n}

Case 1.2. $|L| > 0$. Choose an arbitrary vertex v from L and delete all vertices $L - \{v\}$ from B. The resulting graph will just be a graph of structure $(v \cup C_{2n})^2$

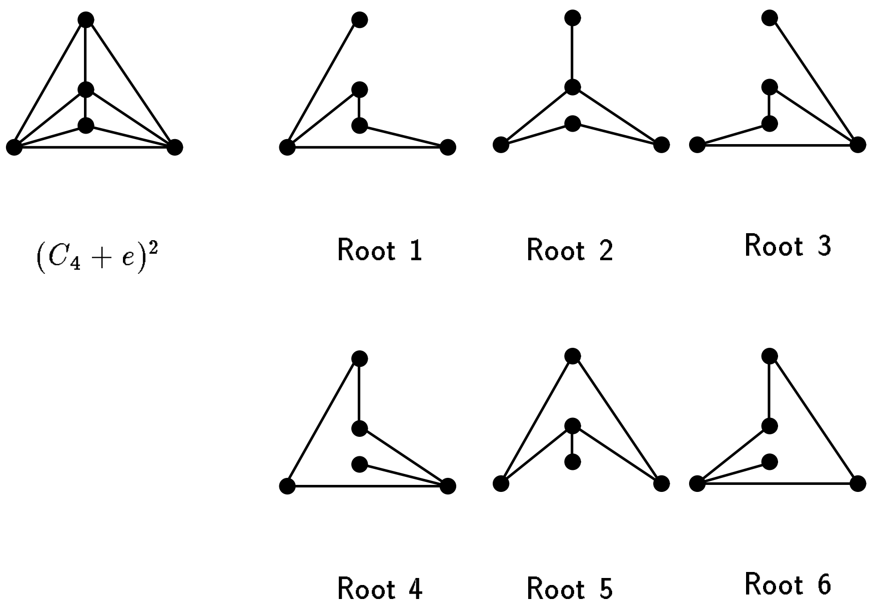

Figure 3: $(C_4 + e)^2$ and its six different roots.

for some $n \geq 2$. Since we know that the vertex v is its leaf, one of vertices $N_B(v) = \{a, b, c\}$ is adjacent to v in B. It is easy to verify that $av \in B$ if and only if $|N_{B^2}(a) \cap N_{B^2}(b)| = |N_{B^2}(a) \cap N_{B^2}(c)| = 3$. Say $av \in B$, meaning ab and $ac \in B$. Since for $x \neq v$ or c and $x \in N_{B^2}(b), bx \in B$ if and only if $ax \in B$, thus once we know $ab, ac \in B$, we can trace the cycle out by this scheme. This process can be repeated until the whole structure of B is revealed. It can be done in $O(|B|)$.

Case 2. $|B| = 5$. For this finite graph, all possible structures of B are

Case 2.1. $B =$ (C_4 plus an endline). This case there are 6 possible arrangements of vertices to constitute B as shown in Figure 3.

Case 2.2. $B =$ (K_3 plus two endlines). Again, there are 6 possible arrangements of vertices to constitute B as shown in Figure 4.

One thing interesting about this case is that there is only one possible (up to isomorphism) square burr for size five. We have shown it has 12 distinct labeled roots and 2 different (up to isomorphic) roots.

Case 3. $|B| = 4$. All possible structures of B are a star with size four and possibly adding any edges or C_4.

Case 4. $|B| = 3$. B is a path of length two or K_3.

Case 5. $|B| \leq 2$. B is K_1 or K_2.

It has been shown that for a large burr, the structure of B can be revealed in $O(|B|)$ and for the finite size of B the structure of B can be found in constant time. We conclude the structure of B can be determined in $O(|B|)$ time. ∎

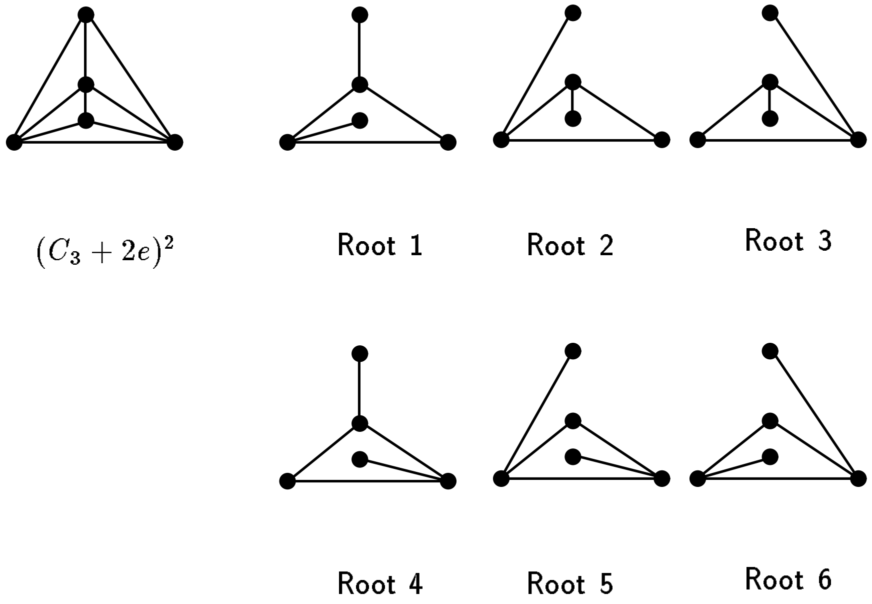

Figure 4: $(C_3 + 2e)^2$ and its six different roots.

Given a square burr B^2, we have just shown that the structure of B is not necessarily unique. Also note that not all combinations of arbitrary burrs roots constitute the root of G^2. So to complete the determination of square root of G^2, we must identify all inner bridges of G. The following lemma shows that it can also be done efficiently.

Lemma 3.6 *Given G^2 a planar graph, all inner bridges of G can be found within $O(n)$ time.*

Proof. Without loss of generality, we assume that G is connected. Say G^2 consists of k square burrs, $B_i^2, 1 \leq i \leq k$. By Lemma 3.3 and Theorem 3.2, these square burrs can be identified in $O(n)$ time. If $k = 1$, then there is no inner bridge in G. We will now only consider the case that $k > 1$.

Let $\mathcal{B} = \{B_i^2 : B_i \text{ is a burr of } G\}$ and $\mathcal{E} = \{(B_i^2, B_j^2) : |B_i^2 \cap B_j^2| = 2\}$. We claim $(\mathcal{B}, \mathcal{E})$ defines a tree. Note that $(\mathcal{B}, \mathcal{E})$ is connected. Suppose there exists a cycle $(B_1^2, B_2^2, ..., B_m^2)$ in $(\mathcal{B}, \mathcal{E})$ where m is the length of the cycle . Then $\cup_{1 \leq i \leq m} B_i$ is a burr, which violates the assumption that B_i is a burr. Therefore $(\mathcal{B}, \mathcal{E})$ must be a tree. It implies that $k \leq n$.

Recall that, for each two burrs B_i and B_j of G, either $B_i^2 \cap B_j^2 = \emptyset$ or $B_i^2 \cap B_j^2 = \{u, v\}$ which implies uv is the inner bridge between B_i^2 and B_j^2. Otherwise, $|B_i^2 \cap B_j^2| = 1$ and there is an intermediate burr B_l^2, which is either a K_3 or K_4, such that $|B_i^2 \cap B_l^2| = |B_l^2 \cap B_j^2| = 2$. Now, repeat the following process for each B_i^2:

First, for each burr B_i^2, insert the number i to each of its vertices, v. Note that each vertex can be associated with at most 4 burrs since the degree of each vertex in the root can be at most 3. Denote the list of burrs associated with vertex v by A_v.

Now, for each vertex v, if the number of square burrs associated with v is at least 2, visit every vertex u in the neighborhood of v. If $|\{A_u \cap A_v\} = \{i, j\}| = 2$, then we can claim that uv is the inner bridge between two burrs B_i and B_j.

The first step of the strategy just mentioned can be done in $O(|B_i|)$ for each B_i^2. Recall that, for each v in G, $|A_v| \leq 4$. Thus the second step can be done in constant time for each vertex v. Totally, we will need $\sum_{1\leq i\leq k}|B_i|+\sum_v 1$, which is exactly $O(n+2k-2)+O(n) = O(n)$ operations (remember that $(\mathcal{B},\mathcal{E})$ is a tree). This completes our proof. ▮

Theorem 3.7 *The square root of a planar graph G^2 can be found in $O(n)$ time.*

Proof. By Lemma 3.3, the square burrs of G^2 can be found in $O(n)$ time. By Lemma 3.6, the inner connection between those square burrs can be found in $O(n)$ time. By Lemma 3.5, the structure of each burr B_i can be found in $O(|B_i|)$ time. Since we has shown in Lemma 3.6 that $\sum_i |B_i| < 3n$, the result follows. ▮

4 Inversion of Total Graphs

Given a graph $G = (V, E)$, its *total graph* [15], $T(G)$, has vertex set $V \cup E$ with two vertices of $T(G)$ adjacent whenever they are neighbors in G. If uv is an edge of G, and w not a vertex of G, then uv is *subdivided* when it is replaced by two edges uw and wv. If every edge of G is subdivided, the resulting graph is the *subdivision graph* $S(G)$ [15]. Behzad [2] showed that, given a graph G, the total graph $T(G)$ is isomorphic to the square of subdivision graph $S(G)$. Behzad and Radjavi [3] showed that the two graphs G, H are isomorphic to each other iff $T(G) \cong T(H)$. We give a polynomial time algorithm for the inversion of total graphs by reducing it into finding square roots of the squares of subdivision graphs. Recently, we have learned of a similar result by Gavril [12].

Given a graph $G = (V, E)$ and its subdivision graph $S(G)$, we call the set of newly added vertices W, such that $V(S(G)) = V \cup W$. By Behzad's result, we know that $T(G) = [S(G)]^2$. Given $v \in V$, we call $I(v) = N_{S(G)}(v) \subset W$, the *inner neighborhoods* of vertex v, and $O(v) = N_G(v) \subset V$, the *outer neighborhoods.* Note that $N_{T(G)}(v) = N_{[S(G)]^2}(v) = I(v) \cup O(v)$. We observe that

Lemma 4.1 *Let uv be an edge of G and w the subdivided vertex of uv implying $uw, wv \in S(G)$. Then*

(i) $I(v) = N_{T(G)}(v) \cap N_{T(G)}(w) - \{u\}$

(ii) For each pair $(w', u') \in I(v) \times O(v), w'u' \in S(G)$ if and only if $\{u'\} = N_{T(G)}(w') \cap N_{T(G)}(v) - I(v)$

Proof. (i) Since, for each $x \in O(v) - \{u\}, d_{S(G)}(w, x) = 3$, it follows that $I(v) = N_{T(G)}(v) \cap N_{T(G)}(w) - \{u\}$ as illustrated in Figure 5.

(ii) The only if part is trivial by (i). Now assume that $\{u'\} = N_{T(G)}(w') \cap N_{T(G)}(v) - I(v)$. Note that $d_{S(G)}(u', w') \leq 2$ and $d_{S(G)}(u', v) = 2$. It follows that $u'v' \in S(G)$. ▮

Given a graph G, recall that a *maximal clique* of G is an induced complete subgraph of G, such that no other vertex in G can be added in to form a larger clique. We find a way to tell whether an edge uv is in the square root graph by examining these maximal cliques in the square containing both vertices u and v.

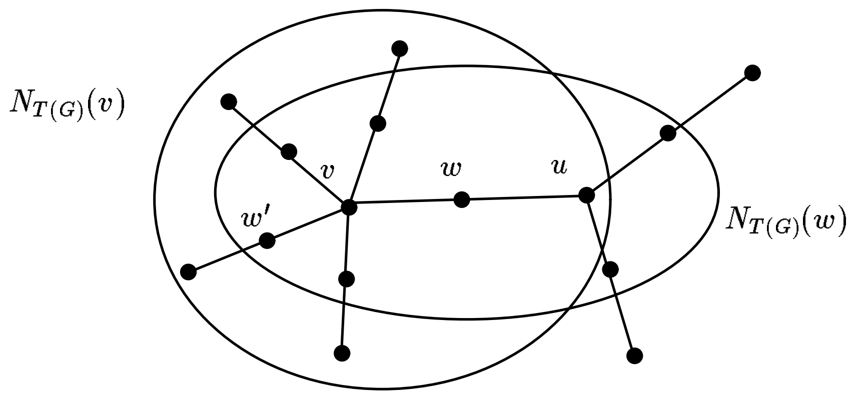

Figure 5: $I(v) = N_{T(G)}(v) \cap N_{T(G)}(w) - \{u\}$ in total graph.

Proposition 4.2 *Given a graph $G = (V, E)$ and an edge uv of G, there are at most two maximal cliques of size 3 containing both u and v in G^2.*

Proof. Let $K = \{u, v, w\}$ be a maximal clique in G^2. Note that it is not possible that $d_G(u, w) = d_G(v, w) = 2$ in G since which will imply that K is not maximal. So, without loss of generality, we can assume that uw is an edge of G. Note that v and w are the only vertices allowed to be adjacent to u since, otherwise, K will not be maximal. It is possible that v can be adjacent to a vertex $w' \neq w$ such that $N_G(v) = \{u, w'\}$ implying $\{u, v, w'\}$ also being a maximal clique, but that is all possible maximal cliques of size 3 we can have containing both u and v. ∎

Proposition 4.3 *Given a graph $G = (V, E)$, an edge uv of G, and w the subdivided vertex of uv in $S(G)$, then $deg_{T(G)}(v) = 2$, and $deg_{T(G)}(w) = deg_{T(G)}(u) + deg_{T(G)}(u)$ both u and v in G^2.*

Proof. The proof of this proposition is based on the fact that $T(G) = [S(G)]^2$, and we will leave it to the reader. ∎

Theorem 4.4 *The inversion of total graphs can be done in $O(m^2)$ time where n and m denote the number of vertices and edges of the given graph respectively.*

Proof. Given a graph H, our goal here is to present an algorithm for finding a graph G such that $H = T(G) = [S(G)]^2$ in $O(|E(H)|^2)$ time. Without loss of the generality, we will assume H, and thus G, is connected. Our strategy is to pick a triple$\{u, v, w\}$ in H such that vw and wu are in $S(G)$, so that we can use Lemma 4.1 to trace the structure of $S(G)$, which is the square root of H, and then the graph G is easily found from $S(G)$.

Assume that H contains n vertices and m edges. First we can construct the adjacency matrix of H in $O(n^2)$ time. Now consider the graph G we are looking for. By Proposition 4.3, if a vertex v has the lowest degree in G, so does v in $T(G)$, i.e., the given graph H. So we begin by sorting all vertices of H according to their degrees in H. Since the maximum degree of all vertices in H is bounded by n, by using the bucket sort method, the sorting process can be done in $O(m)$. Denote the lowest degree of all vertices in H by δ. It is clear that some vertex v in H with degree δ will be also a vertex of G (with degree $\delta/2$,) and there is at least a vertex $w \in N_{T(G)}(v)$ such that vw is also in $S(G)$.

Let α denote the number of degree δ vertices in $T(G)$. It is clear that the number of possible candidates of vw in $T(G)$ will be $\alpha\delta = O(m)$. To apply the result of Lemma 4.1, we also need to find a vertex u in $T(G)$ such that vw and wu are all in G. Fortunately, Proposition 4.3 assures us that the number of candidates of such u is at most two. To find all maximal cliques of size 3 containing vw in $T(G)$, we can examine the set $A = N_{T(G)}(v) \cap N_{T(G)}(w)$ in $O(\delta)$ time, and eliminates any pair $\{x, y\} \subset A$ from it resulting a smaller subset A' if $xy \in T(G)$, which can be done in $O(\delta^2)$ time, which is still bounded by $O(m)$. Again, note that the size of A' is at most two by Proposition 4.3.

Now we have a triple $\{u, v, w\}$ forming a maximal clique in $T(G)$ and suggesting vw and wu being two edges of $S(G)$. So we can apply Lemma 4.1 to trace the structure of $S(G)$ in $O(m+n)$ time by a bread-first-searching method described as following. First, for each vertices w' in the inner neighborhoods, $I(v)$, which can be found in $deg_{T(G)}(v)$ time, we know that $vw' \in E(S(G))$ by part (i) of Lemma 4.1. Also, for each $u' \in N_{T(G)}(w')$, we know that $u'w' \in E(S(G))$ if u' is also in $N_{T(G)}(v)$ but not in $I(v)$, which we can tell in $O(deg_{T(G)}(w'))$ time. By propagating this procedure to each vertex in $T(G)$ by using breadth-first search method, totally, it takes $O(\Sigma_{v\in T(G)} O(deg_{T(G)}(v)) = O(m)$ time.

To avoid the situation that we had made a wrong guess of the $\{u, v, w\}$ triple, we will need to double check whether $H = [S(G)]^2$. This can be done by examining each vertex v of $S(G)$ and see whether $\bigcup_{u\in N_{S(G)}} N_{S(G)}(u) \cup N_{S(G)}(v) = N_H(v)$ in $O(deg_H(v))$ time, since the time needed for finding $\bigcup_{u\in N_{S(G)}} N_{S(G)}(u)$ is $2deg_G(v)$ if $v \in G$, or $deg_G(x) + deg_G(y)$ if v is the subdivided vertex of an edge xy of G, as shown in Proposition 4.3.

Now we have completed the analysis and description of the algorithm. In summary, it will spend at most $O(m)$ time for each possible triple $\{u, v, w\}$, while the number of candidate triples is bounded by $O(\alpha\delta)$. Since we need the time to initialize the adjacency matrix of H, the total time bound will be $O(n^2 + \alpha\delta m)$. In the worst case, it will take $O(m^2)$ time, although in most of the cases that $\alpha\delta$ will be much smaller than m thus the time complexity will collapse into $O(n^2)$. ∎

5 Optimization Problems on Power Graphs

Here we present a number of our results concerning the square, and more generally the powers, of graphs. In Section 5.1, we give a linear time algorithm for finding a Hamiltonian cycle in cubic graphs. In Section 5.2, we prove the NP-completeness of finding the maximum cliques in powers of graphs by transformations from the general maximum cliques problem. In Section 5.6, we prove the chordality of powers of trees by showing that they have the intersection model of subtrees of a tree.

5.1 Hamiltonian Cycles in Powers of Graphs

Fleischner [8] proved that the square of a biconnected graph is always Hamiltonian. Harary and Schwenk [17] proved that the square of a tree T is Hamiltonian if and only if

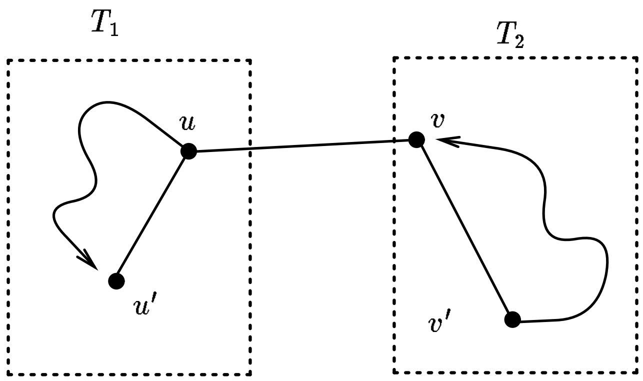

Figure 6: Finding a Hamiltonian cycle in a cubic tree.

T does not contain $S(K_{1,3})$ as its induced subgraph. Here $S(K_{1,3})$ denotes the subdivision graph of the complete bipartite graph $K_{1,3}$, or the size 4 star.

A graph G is *Hamiltonian connected* if every two distinct vertices are connected by a Hamiltonian path. Sekanina [24] proved G^3 is Hamiltonian connected if G is connected. It implies that if the size of G is greater or equal to 3, then G^3 is Hamiltonian. However, we have found no analysis of the complexity of finding Hamiltonian cycles in cubic graphs in the literature. Here we present an linear time (in terms of the vertices and edges number of G) for finding a Hamiltonian cycle in G^3 if G is given.

Theorem 5.1 *Given a connected graph $G = (V, E)$ with size at least 3 and an integer $k \geq 3$, we can find a Hamiltonian cycle in G^k within $O(|V| + |E|)$ time.*

Proof. We will prove this theorem by giving a linear time algorithm for finding a Hamiltonian cycle of G^k. First, we will find a spanning tree, T, of G, while linear time suffices by using a simple depth first search algorithm, assuming the given graph is represented in form of adjacency lists. Our goal is to find a Hamiltonian path h in T^3; it follows that $uv + h$ will be a Hamiltonian cycle of T^3 thus a Hamiltonian cycle of G^3. Since $E(G^i) \subset E(G^{i+1})$ for any positive integer i, $uv + h$ is also a Hamiltonian cycle of G^k.

We start by selecting an arbitrary edge uv of T. Deleting uv from T, separate T into two smaller trees, T_1 and T_2, containing u and v respectively, as shown in Figure 6.

If u is also adjacent to some vertex u' in T_1, we proceed to find a Hamiltonian path P_1 from u to u' in T_1, by recursively applying the algorithm; otherwise, u itself is the Hamiltonian path we want. The same procedure will be applied in T_2; only this time we construct a Hamiltonian path P_2 from v' (a vertex adjacent to v in T_2) to v. Since the last vertex of P_1 (u' or u) has distance at most 3 from the first vertex of P_2 (v' or v), it follows that P_1 concatenating with P_2 is a Hamiltonian path from u to v in T^3.

To demonstrate this algorithm can be done in linear time, we must show that the deletion of uv from T can be implemented in constant time. Note that the conventional adjacent lists representation might take $O(n)$ to delete an arbitrary edge, thus resulting an $O(n^2)$ algorithm. Therefore, We use a doubly linked list $N(v)$, forming a ring, to represent the set of all neighbors of each vertex v in T. Each element of $N(v)$ has a pointer points to an edge $e = uv$, for some u. For each vertex v of T, we use a cell containing a pointer pointing to the double linked list $N(v)$. For each edge $e = uv$, we use four pointers pointing to $u, v, N(u)$, and $N(v)$. Thus, to delete an edge $e = uv$, we start from e, following its link to locate the position of e in $N(v)$ and delete it in constant time; then locate v and link it to any element of this newly modified $N(v)$. The part of u and $N(u)$ can be handled symmetrically.

Let $|T| = |V| = n, |T_1| = n_1$, and $|T_2| = n_2$, and let $time(T)$ define the time needed to perform our algorithm. Since $n = n_1 + n_2$ and

$$time(T) = time(T_1) + time(T_2) + \text{constant time for deleting } uv,$$

by induction, $time(T) = time(T_1) + time(T_2) + c = O(n_1) + O(n_2) + c = O(n)$. So we now conclude that finding a Hamiltonian cycle in a cubic (or higher powers) graph can be done in linear time. ∎

5.2 Maximum Cliques of Power Graphs

Recall that a clique in a graph G is *maximum* if it is the largest induced complete subgraph of G. Here we prove that finding maximum cliques in powered graphs is NP-hard by transformation from the general problem of finding the maximum cliques in arbitrary graphs.

Theorem 5.2 *Let $G = (V, E)$ be a graph. Then, for any fixed integer $k \geq 1$, finding the maximum clique of G^k is NP-complete.*

Proof. Note that, for the case of $k = 1$, this problem is to find the maximum clique in G, which is NP-complete as shown in [10]. Let $|E(G)| = m$. Now we will prove the theorem for the case that $k > 1$. We divide the problem depending upon whether k is even or odd:

Case 1: $k = 2p + 2$, for some $p \geq 0$. Construct a graph G' according to the following transformation. For each edge $uv \in E$, construct a set of $2p + 1$ new vertices, $V_{uv} = \{[uv]_0, [uv]_1, ..., [uv]_{2p}\}$. Let $W = \cup_{uv \in E} V_{uv}$, the new vertices in G'. Link these vertices of V_{uv} with u and v to form a path by adding the edges $E_{uv} = \{([uv]_i, [uv]_{i+1}) : 0 \leq i < 2p\} \cup \{(u, [uv]_0), ([uv]_{2p}, v)\}$. Note that $[uv]_p$ is the unique center vertex in E_{uv}. Link these center vertices in G' by defining $C = \{([uv]_p, [u'v']_p) : \text{ for all } uv, u'v' \in E\}$. Figure 7 illustrates this transformation.

Now construct the graph $G' = (V', E')$ by defining

$$\begin{aligned} V' &= V \cup W \\ E' &= \cup_{uv \in E} E_{uv} \cup C \end{aligned}$$

Since, for each pair of vertices $x \in V_{uv}, y \in V_{u'v'}$, the path from x to $[uv]_p$, to $[u'v']_p$, and, finally, to y has length at most $2p + 1 = k - 1$. It follows that, for each pair $v \in V$ and $w \in W$, $d_{G'}(v, w) \leq k$, implying that $W \cup v$ forms a clique in G'^k for any $v \in V$. Also notice that, for each two vertices, u and v, of V, $uv \in E$ if and only if $d_{G'}(u, v) = k$. Thus, a subset of V, Q of size q, is a clique of G if and only if $X \cup W$ is a clique of $(G')^k$. That is, G has a clique of size q if and only if G'^k has a clique of size $(k - 1)m + q)$. Since

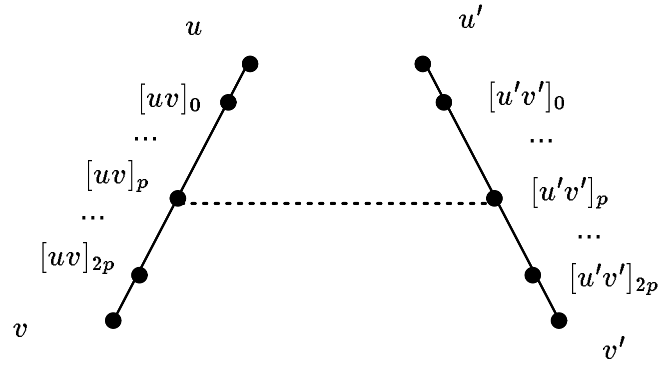

Figure 7: The transformation of even powers of a graph.

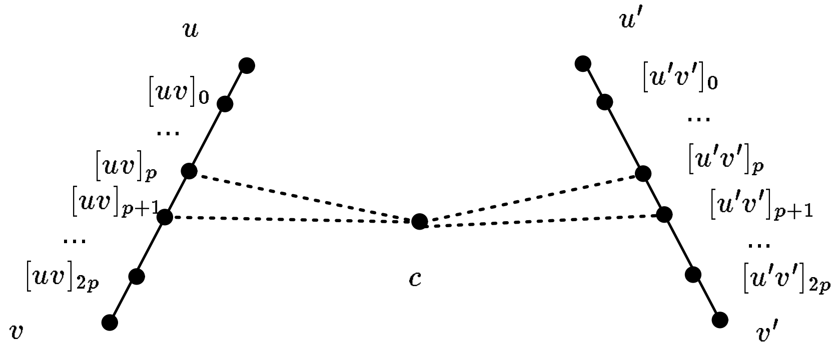

Figure 8: The transformation of odd powers of a graph.

clearly G' can be construct in polynomial time, thus finding the maximum clique in G^k is NP-hard for the case of even k.

Case 2: $k = 2p + 1$, for some $p \geq 1$. The construct of the graph G' is similar to the even case except that we now add an additional kernel vertex c to G'. For each edge $uv \in E$, construct a set of $2p$ new vertices, $V_{uv} = \{[uv]_1, [uv]_1, ..., [uv]_{2p}\}$. The new vertices in G' will be $W = \cup_{uv \in E} V_{uv} \cup \{c\}$. Link vertices of V_{uv} with u and v to form a path. by adding the edges $E_{uv} = \{([uv]_i, [uv]_{i+1}) : 1 \leq i < 2p\} \cup \{(u, [uv]_0), ([uv]_{2p}, v)\}$. Note that $[uv]_p$ and $[uv]_{p+1}$ are two center vertices in E_{uv}. Link these center vertices with the kernel c by defining $C = \{(c, [uv]_i) : \text{ for all } uv \in E, i \in [p, p+1]\}$. Figure 8 illustrates this transformation.

Again, construct the graph $G' = (V', E')$ by defining

$$\begin{aligned} V' &= V \cup W \\ E' &= \cup_{uv \in E} E_{uv} \cup C \end{aligned}$$

Since, for each pair of vertices $x \in V_{uv}, y \in V_{u'v'}$, the path from x to y defined by concatenating the shortest path from x to c and from c to y has a length at most $2p = k-1$, it follows that, for each pair $v \in V$ and $w \in W$, $d_{G'}(v, w) \leq k$, implying that $W \cup v$ forms a clique in G'^k for any $v \in V$. Also notice that, for each two vertices, u and v, of V, $uv \in E$ if and only if $d_{G'}(u, v) = k$. Thus, given X, a subset of V, X is a clique of G

if and only if $X \cup W$ is a clique of $(G')^k$. That is, G has a clique of size q if and only if G'^k has a clique of size $(k-1)m+1+q)$. It is not difficult to verify that G' can be constructed in polynomial time; thus, the problem is NP-hard for the case of odd k.

Since the problem is clearly in NP, we now conclude the problem of finding the maximum clique of G^k is NP-complete for any fixed integer $k \geq 1$. ∎

5.3 Maximum Independent Sets of Power Graphs

Given a graph G and a positive integer $k > 1$, the transformed graph G' described in the previous proof is uniquely defined. We can denote this transformation by: $G' = \Phi(G,k)$. The most interesting property of this transformation is that, for every two distinct vertices u, v of G, we have:

$$uv \notin G \iff d_{\Phi(G,k)}(u,v) = k+1 \tag{1}$$

and

$$uv \in G \iff d_{\Phi(G,k)}(u,v) = k. \tag{2}$$

Further, for every two new vertices $x, y \in W$ of $\Phi(G,k)$, and each original vertex v of G, we have:

$$d_{\Phi(G,k)}(v,x) \leq k \quad and \quad d_{\Phi(G,k)}(x,y) < k \tag{3}$$

Thus, $\Phi(G,k)^k$, the k-th power of the transformed graph $\Phi(G,k)$, still contains the information of the original clique. That is, assuming $Q \subset V(G)$, Q is a clique in G iff Q is a clique in $\Phi(G,k)^k$. This observation essentially constitutes the proof of Theorem 5.2. Interesting enough, we will find that $\Phi(G,k)^k$ also contains information of the original *independent set.* Formally, we have

Theorem 5.3 *Let $G = (V,E)$ be a graph. Then, for any fixed integer $k \geq 1$, finding the maximum independent set of G^k is NP-complete.*

Proof. For the case of $k = 1$, this problem is to find the maximum independent set in G, which is NP-complete as shown in [10]. Otherwise, we can reduce the problem of finding the maximum independent set in a general graph to this problem for any fixed $k > 1$.

Given an arbitrary graph $G = (V,E)$, we can construct the graph $\Phi(G,k)$ in polynomial time as shown in the proof of Theorem 5.2. We will now show that G has an independent set of size α iff $\Phi(G,k)^k$ has an independent set of size α. Let I, $|I| = \alpha$ be an independent set of G. By Equation (1), I is clearly an independent set of $\Phi(G,k)^k$.

On the other direction, let I, $|I| = \alpha$, be an independent set of $\Phi(G,k)^k$. Without loss of generality, we can assume $\alpha > 1$. By Equation (3), I can not contain any new vertex (W); thus $I \subset V$. Again, by Equation (1), I is an independent set of G. ∎

5.4 Coloring and Cliques Partition of Power Graphs

Given a graph G, a *clique partition* of G is a partition $\{V_1, \ldots, V_k\}$ of the vertices of G such that each V_i, $1 \leq i \leq k$, is a clique in G. The *minimum* clique partition of G is a clique partition with the smallest possible k. A *vertex coloring* of G is a partition

$\{V_1, \ldots, V_k\}$ of the vertices of G such that each V_i, $1 \leq i \leq k$, is an independent set in G. The *minimum* vertex coloring of G is a coloring with the smallest possible k. Once agin, the transformation Φ can be used to prove the the hardness of finding a minimum clique partition or finding a minimum coloring of in a power graph:

Theorem 5.4 *Let $G = (V, E)$ be a graph. Then, for any fixed integer $k \geq 1$, finding the minimum clique partition of G^k is NP-complete.*

Proof. For the case of $k = 1$, this problem is to find the minimum clique partition in G, which is NP-complete as shown in [10]. Otherwise, we can reduce the problem of finding the minimum clique partition in general graph to this problem for any fixed $k > 1$.

Given an arbitrary graph $G = (V, E)$, we can construct the graph $\Phi(G, k)$ in polynomial time as shown in the proof of Theorem 5.2. We will now show that G has a clique partition of size $\leq \beta$ iff $\Phi(G, k)^k$ has a clique partition of size $\leq \beta$. Let $P = \{V_1, \ldots, V_\alpha\}$, $\alpha \leq \beta$, be a clique partition of G. By Equation (2), for each $V_i \in P$, $1 \leq i \leq \alpha$, V_i is again a clique in $\Phi(G, k)^k$. Further, by (3), $V_1 \cup W$ is still a clique. Thus $\Phi(G, k)^k$ has a clique partition of size $\alpha \leq \beta$.

On the other direction, let $P = \{V_1, \ldots, V_\alpha\}$, $\alpha \leq \beta$, be a clique partition of $\Phi(G, k)^k$. For each vertices subset V_i, $1 \leq i \leq \alpha$, let $V_i' = V_i \cap V$. For each i, clearly V_i' (if not empty) is still a clique in $\Phi(G, k)^k$. Further, by Equation (2), V_i' is also a clique in G. It follows that $\{V_1', \ldots, V_\alpha'\}$ is a clique partition of G. That is, G has a clique partition of size $\leq \beta$ ∎

Theorem 5.5 *Let $G = (V, E)$ be a graph. Then, for any fixed integer $k \geq 1$, finding the minimum vertex coloring of G^k is NP-complete.*

Proof. For the case of $k = 1$, this problem is to find the minimum vertex coloring in G, which is NP-complete as shown in [10]. Otherwise, we can reduce the problem of finding the minimum vertex coloring in general graph to this problem for any fixed $k > 1$.

Given an arbitrary graph $G = (V, E)$, we can construct the graph $\Phi(G, k)$ in polynomial time as shown in the proof of Theorem 5.2. Let $w = |W|$ denote the size of the newly added vertices of $\Phi(G, k)$. We will now show that G has a vertex coloring of size α iff $\Phi(G, k)^k$ has a vertex coloring of size $\alpha + w$. Let $C = \{V_1, \ldots, V_\alpha\}$ be a vertex coloring of G. By Equation (1), each $V_i \in P$, $1 \leq i \leq \alpha$, is again an independent set in $\Phi(G, k)^k$. For each $v \in W$, singleton $\{v\}$ certainly constitute an independent set. Thus $\{V_1, \ldots, V_\alpha, \{v\}_{v \in W}\}$ is a coloring of size $\alpha + w$.

In the other direction, let $C = \{V_1, \ldots, V_{\alpha+w}\}$ be a vertex coloring of $\Phi(G, k)^k$. By Equation (3), each new vertex $v \in W$ is adjacent to every other vertices in $\Phi(G, k)^k$. Thus each vertex $v \in W$ must form a singleton in the coloring C. Deleting those singletons from C, we have $C' = C \setminus \{\{v\} : v \in W\}$. Note that $|C'| = \alpha$, and C' is a vertex coloring of G since each vertices set of C' is still an independent set in G by Equation (1). That is, G has a vertex coloring of size α. ∎

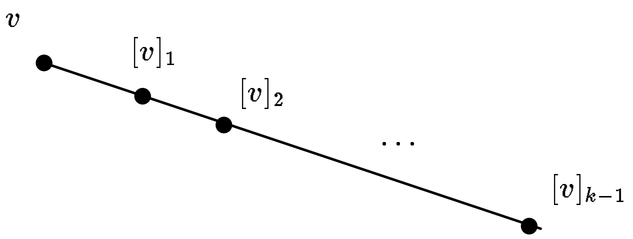

Figure 9: Adding a length $k-1$ tail to a vertex v.

5.5 Minimum Dominating Sets of Power Graphs

Given a graph $G=(V,E)$, a *dominating set*, D, is a subset of vertices of G such that for each vertex $v \in V \setminus D$, there is some vertex $u \in D$ such that $uv \in E$. In other words, $\cup_{v\in D} N^+(v) = V$. A *minimum dominating set* is a dominating set with the smallest cardinality. It would be nice if the same transformation can also be used to prove that the minimum dominating set problem in G^k is also NP-complete. Unfortunately, since the size of the minimum dominating set in $\Phi(G,k)^k$ is always 1, we must use another technique.

Theorem 5.6 *Let $G=(V,E)$ be a graph. Then, for any fixed integer $k \geq 1$, finding the minimum dominating set of G^k is NP-complete.*

Proof. For the case of $k=1$, this problem is to find the minimum dominating set in G, which is NP-complete as shown in [10]. Otherwise, we can reduce the problem of finding the minimum dominating set in general graph to this problem for any fixed $k>1$.

Given k and an arbitrary graph $G=(V,E)$, we will construct the following graph, denoted $\Psi(G,k)$, in polynomial time: For each vertex v of G, we will attach a *tail* of length $k-1$ to it. That is, for each $v \in V$, let

$$\begin{aligned} V_v &= \{[v]_1, \ldots, [v]_{k-1}\} \\ E_v &= \{v[v]_1\} \cup \{[v]_i[v]_{i+1} : i \in [1..k-2]\} \end{aligned}$$

Figure 9 illustrates this transformation. Now we can define $\Psi(G,k) = (V', E')$ as following:

$$\begin{aligned} V' &= \cup_{v\in V} V_v \cup V \\ E' &= \cup_{v\in V} E_v \cup E \end{aligned}$$

Note that $\Psi(G,k)$ has kn vertices and $m+(k-1)n$ edges. $\Psi(G,k)$ can certainly be constructed in polynomial time. Note that graphs G and $\Psi(G,k)$ are related by the following equation: For $v, x \in V, i \in [1..k-1]$,

$$d_{\Psi(G,k)}(v, [x]_i) = d_G(v,x) + i \tag{4}$$

We will now show that G has a dominating set of size α iff $\Psi(G,k)^k$ has a dominating set of size α. Let $D = \{v_1, \ldots, v_\alpha\}$ be a dominating set of G. That is, for each $x \in V' \setminus D$,

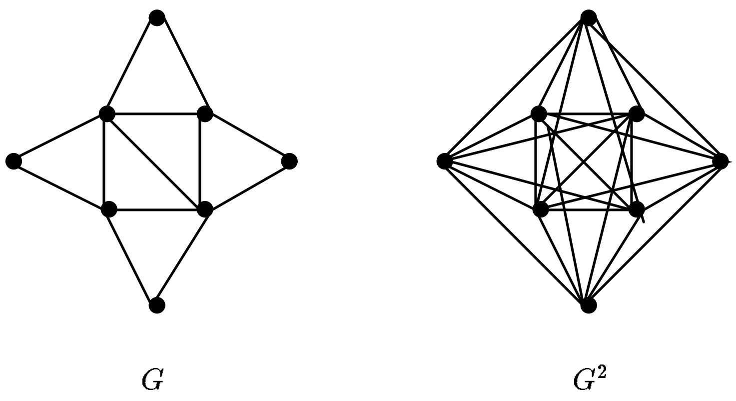

Figure 10: A chordal graph and its nonchordal square.

there exists a $v \in D$ such that $d_G(v, x) = 1$. By Equation (4), for each $[x]_i \in V \setminus D$, $d_G(v, [x]_i) \leq k$ since $i \in [1..k-1]$. That is, D is still a dominating set of $\Psi(G, k)^k$.

In the other direction, let $D = \{v_1, \ldots, v_\alpha\}$ be a dominating set of $\Psi(G, k)^k$. Without loss of generality, we can assume that $D \subset V$. Otherwise, for each $[v]_j \in D$, replace $[v]_j$ by the vertex v. Call the modified set D'. By Equation (4), D' is still a dominating set of $\Psi(G, k)^k$. We claim that D' is also a dominating set of G: Suppose D' is not. Then there must be a vertex v in G such that for each $x \in D'$, $d_G(v, x) > 1$. By Equation (4), that means, for each $x \in D'$, $d_{\Psi(G,k)}(v, x) > k$, but it is contradicted by the fact that D' is a dominating set of $\Psi(G, k)^k$. Since D' is a dominating set of G, and D' has the same cardinality of D, it follows that G has a dominating set of size α. ▮

5.6 The Chordality of Powers of Trees

A graph whose every simple cycle of length strictly greater 3 possesses a chord is called a *chordal graphs.* In the literature, chordal graphs have also been called *triangulated, rigid-circuit, monotone transitive,* and *perfect elimination* graphs [6, 19, 9, 22]. Rose, Tarjan, and Lueker [22] presented an algorithm for recognition of chordal graph in linear time. Along with their algorithm, the maximum clique of a chordal graph can also be found in linear time. Walter [28], Gavril [11], and Buneman [4] proved that a graph G is chordal if and only if G is the intersection graph of a family of subtrees of a tree. Observe that the square of a chordal graph is not necessarily chordal, as shown in Figure 10. However, we will show that any arbitrary powers of trees are always chordal.

Theorem 5.7 *Powers of trees are chordal.*

Proof. Let T be a tree. We want to prove that T^k is chordal for any positive integer k. Let $S(G)$ be the subdivision graph of a graph G as defined in Section 4, and $N_G^k(v)$ denotes the set $\{u \in G : d(u, v) \leq k\}$ where $d(u, v)$ is the distance between vertices u and v in G. Note that $uv \in G^k$ if and only if $N_{S(G)}^k(u) \cap N_{S(G)}^k(v) \neq \emptyset$, implying that G^k is the intersection graph of the family $S = \{N_{S(G)}^k(v) :$ for all $v \in G\}$. It follows that T^k is the intersection graph of the family $\{N_{S(T)}^k(v) :$ for all $v \in T\}$. Since $S(T)$ is a tree and $\langle N^k_{S(T)}(v)\rangle_{S(T)}$ is a subtree of $S(T)$ with center vertex v and radius k, T^k is the intersection graph of a family of subtrees of the tree $S(T)$. Thus T^k is chordal. ∎

6 Conclusions

We have presented efficient algorithms for finding the square roots of graphs in three interesting special cases: tree squares, planar graphs and subdivided graphs. Further, we have studied the complexity of several optimization problems on powers of graphs. Two interesting open problems remain:

- Let A be the adjacency matrix of a graph, made reflexive by adding a self-loop to each vertex. Given A^2, can we determine one of its $(0,1)$-matrix square roots in polynomial time? This problem is potentially easier than the finding the square root of a graph, since we are also given the number of paths of length at most two between each pair of vertices.

- What is the complexity of recognizing the squares of directed-acyclic graphs? Clearly, the square of DAG is a DAG. All square roots of a DAG G^2 contain the transitive reduction of G^2 as a subgraph. The time complexity of finding both the transitive closure and reduction of a digraph is equivalent to boolean matrix multiplication [1].

7 Acknowledgments

We thank Gene Stark for providing a simpler proof of Theorem 5.7, and Alan Tucker for valuable discussions.

References

[1] A. Aho, M. Garey, and J. Ullman. The transitive reduction of a directed graph. *SIAM Journal on Computing*, 1:131–137, 1972.

[2] M. Behzad. A criterion for the planarity of a total graph. *Proc. Cambridge Philos. Soc.*, 63:679–681, 1967.

[3] M. Behzad and H. Radjavi. The total group of a graph. *Proc. Amer. Math. Soc.*, 19:158–163, 1968.

[4] P. Buneman. A characterization of rigid circuit graphs. *Discr. Math.*, 9:205–212, 1974.

[5] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. In *Proc. 19th ACM Symp. Theory of Computing*, pages 1–6, 1987.

[6] G.A. Dirac. On rigid circuit graphs. *Abh. Math. Sem. Univ. Hamburg*, 25:71–76, 1961.

[7] F. Escalante, L. Montejano, and T. Rojano. Characterization of n-path graphs and of graphs having nth root. *J. Combin. Theory B*, 16:282–289, 1974.

[8] H. Fleischner. The square of every two-connected graph is Hamiltonian. *J. Combin. Theory B*, 16:29–34, 1974.

[9] D.R. Fulkerson and O.A. Gross. Incidence matrices and interval graphs. *Pacific J. Math.*, 15:835–855, 1965.

[10] M.R. Garey and D.S. Johnson. *Computers and Intractability – A Guide to the Theory of NP-Completeness.* Freeman, New York, 1979.

[11] F. Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *J. Combin. Theory B*, 16:47–56, 1974.

[12] F. Gavril. A recognition algorithms for the total graph. *Networks*, 8:121–133, 1978.

[13] D.P. Geller. The square root of a digraph. *J. Combin. Theory*, 5:320–321, 1968.

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

[15] F. Harary. *Graph Theory.* Addison-Wesley, Massachusetts, 1972.

[16] F. Harary, R.M. Karp, and W.T. Tutte. A criterion for planarity of the square of a graph. *J. Combin. Theory*, 2:395–405, 1967.

[17] F. Harary and A. Schwenk. Trees with hamiltonian square. *Mathematika*, 18:138–140, 1971.

[18] J.E. Hopcroft and R.E. Tarjan. Dividing a graph into triconnected components. *SIAM Journal on Computing*, 2:135–158, 1973.

[19] C.G. Lekkerkerker and J.Ch. Boland. Representation of a finite graph by a set of intervals on the real line. *Fund. Math.*, 51:45–64, 1962.

[20] R. Motwani and M. Sudan. Square is NP-complete. Manuscript, September 1991.

[21] A. Mukhopadhyay. The square root of a graph. *J. Combin. Theory*, 2:290–295, 1967.

[22] D.J. Rose, R.E. Tarjan, and G.S. Lueker. Algorithmic aspects of vertex elimination of graphs. *SIAM Journal on Computing*, 5:266–283, 1976.

[23] I.C. Ross and F. Harary. The square of a tree. *Bell System Tech. J.*, 39:641–647, 1960.

[24] M. Sekanina. On an ordering of the set of vertices of a connected graph. Technical Report No. 412, Publ. Fac. Sci. Univ. Brno, 1960.

[25] R.E. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1:146–159, 1972.

[26] W.T. Tutte. *Connectivity in Graphs.* University of Toronto Press, London, 1966.

[27] P. Underground. On graphs with Hamiltonian squares. *Discr. Math.*, 21:323, 1978.

[28] J.R. Walter. *Representations of rigid cycle graphs.* PhD thesis, Wayne State Univ., 1972.