Square Root Finding In Graphs

Majid Karimi, Master of Science

Mathematics and Statistics

Submitted in partial fulfilment
of the requirements for the degree of

Master of Science

Faculty of Mathematics and Science, Brock University
St. Catharines, Ontario

Abstract

Root and *root finding* are concepts familiar to most branches of mathematics. In graph theory, H is a *square root* of G and G is the *square* of H if two vertices x, y have an edge in G if and only if x, y are of distance at most two in H. Graph square is a basic operation with a number of results about its properties in the literature. In this thesis, we are interested in the characterization and recognition problems of graph powers. There are algorithmic and computational approaches to answer the decision problem of whether a given graph is a certain power of any graph. On the other hand characterization of graph powers such as the work of Mukhopadhyay [24] do not provide a polynomial time algorithm for the recognition problem.

One of the best known results for the problem is given by Farzad et al. [11]. They approached the problem in terms of the girth of the square root. They provided an almost dichotomy theorem for the problem. However the problem has been open in the case where the root graph has girth five. We settle, in the affirmative, the conjecture that recognition of square of graphs with girth 5 is NP-complete. We also present a polynomial time algorithm for graphs of girth five without specific dense subgraphs.

to my lovely wife,

for all of her support, courage

and love

Acknowledgement

I am indebted to my supervisor Dr. Babak Farzad for his continuous support. This thesis would not have been possible without his encouraging approach, help and notable support. My gratitude is also expressed to the Department of Mathematics and Statistics for supporting me over the period of my Master studies.

I also want to thank my kind wife Niloufar for all those support, patient and understanding. Many thanks to my family and friends for their support.

Majid Karimi

September 2012

Contents

Chapter 1

Introduction

Graph-theoretical ideas date back to at least the 1730's, when Leonhard Euler published his paper on the problem of Seven Bridges of Köningsberg. This puzzle asks whether there is a continuous walk that crosses each of the seven bridges of Köningsberg only once and if so, whether a closed walk can be found, see Figure 1.1.

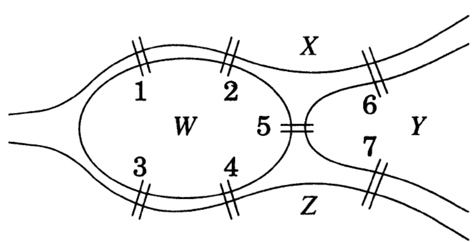

Figure 1.1: Bridges of Köningsberg

The methods for solving such games or problems was the main motivation for a theory called *Graph Theory*.

Graphs are very convenient tools for representing the relationships among objects, which are represented by vertices. Also relationships among vertices are represented by connections. In general, any mathematical object involving points and connections among them can be called a graph or a hypergraph. For a great diversity of problems such pictorial representations may lead to a solution.

Nowadays, graph theory is a dynamic field in both theory and applications. Graphs can be used as a modelling tool for many problems of practical importance. For instance, a network of cities, which are represented by vertices, and connections among them make a weighted graph. The link structure of a website could be represented by a directed graph. The vertices are the web pages available at the website and a directed edge from page A to page B exists if and only if A contains a link to B. A similar approach can be taken to problems in travel, biology, computer chip design, and many other fields.

Root finding problem in graphs has been extensively studied in the past three decades. As a very basic operation in algebra, we may refer to the n^{th} power of a number x^n, and the n^{th} root of a number $\sqrt[n]{x}$. The same attempt to such operations in graph theory was first introduced by Harary and Ross in [27]. They defined the square of a tree T denoted by T^2, to be graph in which two vertices x, y have an edge if and only if x, y are of distance at most two in T. However the same definition work for an arbitrary graph H.

For a given graph H, computing the square graph H^2 is an easy operation. However for a given graph G deciding if there exists a graph H such that $G = H^2$ is not that easy.

Ross and Harary [27] characterized squares of trees and showed that tree square roots, when they exist, are unique up to isomorphism. Mukhopadhyay [24] provided a characterization of graphs which have a square root. However his characterization does not give a short certificate. In fact, such a good characterization may not exist as Motwani and Sudan [23] proved that it is NP-complete to determine if a given graph has a square root.

Meanwhile, there are polynomial time algorithms to compute the tree square root [21, 18, 19, 6, 7], a bipartite graph square root [19], and a proper interval graph square root [20].

As an *almost* complete dichotomy theorem in square root finding problem, Farzad et al. [11] provided the fastest known algorithm for the recognition of square of graphs with girth at least 6. They also proved the NP-completeness of the square root finding problem for graphs with girth four. Putting all results together Table 3.1 is representing the square root problem regarding the girth of square root (indicated by H). The recognition problem has been open for square roots of girth 5.

Girth	Complexity Class
∞	P [7]
$g(H) \geq 7$	P [11]
$g(H) \geq 6$	P [11]
$g(H) \geq 5$	?
$g(H) \geq 4$	NP-Complete [11]
$g(H) \geq 3$	NP-Complete [23]

Table 1.1: Known complexity classes for Square Root Problem.

1.1 Overview

In addition to introducing some preliminaries and notations in Chapter 2, we introduce the concept of graph powers and decision problem of root finding in Section 2.2. In Chapter 3, we present the previous works on the complexity problem of root finding with a focus on locality. We present the result of Farzad et al. [11] for square of graph with girth at least 6. We also present the NP-completeness proof of Motwani and Sudan [23] for finding square roots of graphs in general and the NP-completeness proof of Farzad et al. [11] for finding square roots of girth 4 of graph. We also discuss the uniqueness results for square roots in terms of their girth.

In Chapter 4 we settle, in the affirmative, the conjecture that recognition of square of graphs with girth 5 is NP-complete. We also present a polynomial time algorithm for graphs of girth 5 without specific dense subgraphs.

Chapter 5 contains the concluding remarks and discussions about possible further directions.

Chapter 2

Preliminaries and Notions

In this chapter we introduce graph-theoretical definitions and terminologies for the remainder of this thesis. For undefined notations and definitions that are not included in this chapter, we use [30] as our reference.

2.1 Preliminaries and Notions

A graph G is an ordered pair $G = (V, E)$ where V is a set of elements of which are called *vertices* or nodes, and E is a set of unordered pairs of distinct vertices called *edges* or lines. Each vertex of an edge e is called an *endpoint* of e and we show $e = vu$ for v and u two different endpoints of e. When u and v are the endpoints of an edge, they are *adjacent* and are *neighbours*. We write $u \sim v$ for u is adjacent to v. A finite graph is a graph such that $V(G)$ and $E(G)$ are finite sets. All graphs studied in this thesis are finite and undirected.

An *isomorphism* from a graph G to a graph H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ if and only if $f(u)f(v) \in E(H)$. We say G is *isomorphic* to H, written $G \cong H$, if there is an isomorphism from G to H. Since H is isomorphic to G whenever G is isomorphic to H, we often say G and H are isomorphic. It is an observation that the isomorphism relation is an equivalence relation on the set of simple graphs. An *isomorphism class* of graphs is an equivalence class of graphs under the isomorphism relation. The *adjacency matrix* of a graph G on n vertices $\{v_1, v_2, ..., v_n\}$, written $A(G)$, is the $n \times n$ matrix in which entry $a_{i,j}$ is 1, if there exists an edge between v_i and v_j, and 0 otherwise.

If G is a graph of order n, then $V(G) = \{v_1, v_2, \dots, v_n\}$ is the set of vertices in G and $E(G) = \{e_1, e_2, \dots, e_m\}$ is the set of edges. Also, $d(v_i)$ denotes the degree of v_i. In a graph G denote by $\delta(G)$ and $\Delta(G)$ the *minimum* and *maximum* degrees of G, respectively. A k-regular graph is graph in which every vertex has a degree of k. In a graph G, a *walk* from vertex v_i to vertex v_j is an alternating sequence of vertices and edge $W = \langle v_i, \underbrace{v_i v_{i+1}}_{e_1}, v_{i+1}, \dots, \underbrace{v_{j-1} v_j}_{e_k}, v_j \rangle$. The number of edges in W is called the *length* of W. The distance from a vertex v to a vertex u is the length of the shortest walk from v to u. We may use $d_G(v, u)$ or simply $d(v, u)$ for the distance notation between v and u in graph G. Also if there is no walk from v to u then $d(u, v) = \infty$. The *diameter* of a graph G is defined as maximum of distances in G, or $max_{u,v \in V(G)} d_G(u, v)$. A graph G is said to be *connected* if there exists a path between every pair of vertices and *disconnected* otherwise. The diameter of a disconnected graph is defined to be ∞.

For a vertex v in graph G, the set of all vertices adjacent to v is called the *neighbourhood* of v showed by $N_G(v)$. The closed neighbourhood of a vertex v is its neighbourhood and v itself and we show it by $N_G[v]$ or simply $N[v]$. In other words, $N[v] = N(v) \cup \{v\}$.

A *path* is a walk without vertex repetition. A *cycle* is a closed path meaning the first and the last vertex of the path are the same. The length of the shortest cycle in the graph G is called the *girth* of the graph and is shown by $g(H)$.

A subgraph of a graph $G = (V, E)$, is another graph $G' = (V', E')$ such that $V' \subseteq V$ and $E' \subseteq E$. For S a subset of vertices, we write $G - S$ for the subgraph obtained by deleting S (and of course edges having an endpoint in S). An *induced subgraph* of G, is a subgraph obtained by deleting a set of vertices. We write $G[T]$ for $G - \bar{T}$ such that $\bar{T} = V(G) - T$, which is the subgraph of G *induced by* T. A graph G is called H-free if G has no induced subgraph isomorphic to H.

A graph on n vertices is called *complete* when every vertex is adjacent to all other vertices. Such a graph is denoted by K_n. A complete subgraph of a graph G is called a *clique* of G. The size of a maximum clique in G is called the *clique number* of G.

Definition 1 (Petersen Graph)**.** *The Petersen graph is the simple graph whose vertices are the* 2*-element subsets of a* 5*-element set and whose edges are all pairs of disjoint two element subsets, see Figure 2.1.*

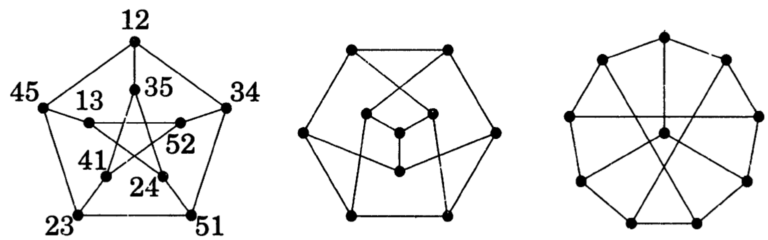

Figure 2.1: Different drawings of Petersen graph

A Moore graph is a k-regular graph with diameter d, with $k^2 + 1$ vertices. An equivalent definition of a Moore graph is that it is a graph of diameter d with girth $2d + 1$. The Petersen graph also can be considered as a Moore graph of diameter 2, and girth $2 \times 2 + 1 = 5$, see Figure 2.2.

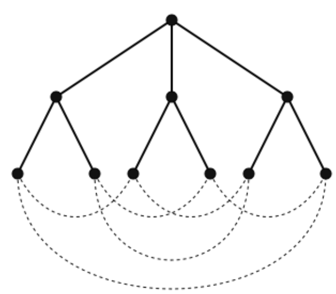

Figure 2.2: Petersen graph as Moore graph

It also can be seen that the maximum number of vertices in a k-regular graph of diameter d with girth $2d + 1$ is $k^2 + 1$ [30].

2.2 Graph Powers, Graph Roots

In this section we introduce the cardinal concepts of this thesis which are *graph power* and *graph root*.

Definition 2. *Graph G is called the r^{th} power of H and H is called the r^{th} root of G where for every pair of vertices v and u, v is adjacent to u in G if and only if $d_H(v,u) \leq r$.*

Example 1. *The graph shown in Figure 2.3 is an example of a graph and its square and cubic graph. Notice that the only pair of vertices of distance four in H are the only non-adjacent ones in H^3.*

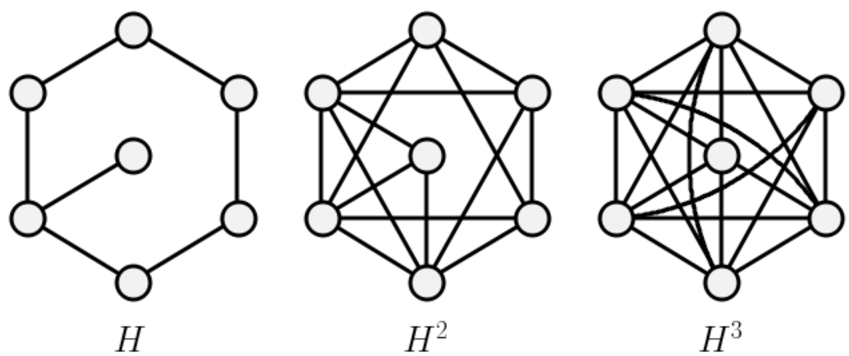

Figure 2.3: Example of Power Graph.

Proposition 1. *Let G be the square of the graph H, then $A(G) = A(H)^2 + A(H)$, where addition is defined to be the binary operation of or*[1]*.*

[1] $x + y = 0$ iff $x = y = 0$.

As a very trivial extension to Proposition 1, for G the n^{th} power graph of H:

$$A(G) = \Sigma_{k=1}^{r} A(H)^k$$

However the other direction of this problem is not trivial. Indeed for given G and r, determining if there exists H such that H is the r^{th} root of G is not an easy problem. Considering the square root of a graph we have the following theorem by Mukhopadhyay [24].

Theorem 1 (Mukhopadhyay)**.** *A connected graph with n vertices $v_1, v_2, \ldots, v_n$ has a square root if and only if some set of n cliques of G whose union is G can be labeled $C_1, C_2, \ldots, C_n$ so that, for all $1 \leq i, j \leq n$ the following conditions hold:*

(i) C_i contains v_i;

(ii) C_i contains v_j if and only if C_j contains v_i.

A detailed proof of Proposition 1 deals with the *maximum clique* problem.

Clique Problem

Instance:	A graph G, a positive integer k.
Question:	Does there exist a clique of size k in G.

The clique problem is an *NP-complete* problem (one of the Karp's 21 NP-complete problems [31]). That means any given solution to such a problem can be verified quickly, however there is no known efficient algorithm to find a solution for an arbitrary instance.

So the problem of finding a maximum clique is NP-complete if one could solve it, one could also solve the decision problem, by comparing the size of the maximum clique to the size parameter given as input in the CLIQUE problem.

Hence Mukhopadhyay theorem is not interesting from a complexity point of view. So the complexity problem of root finding problem is interesting.

Definition 3 (Root Problem)**.** *We call the following decision problem r^{th}* ROOT PROBLEM*:*

CLIQUE PROBLEM

Instance: *A graph G.*

Question: *Does there exist a graph H such that $G = H^r$?*

A good deal of this study is dedicated to solving this problem when $r = 2$ i.e. the problem of square root finding. The reason is that a dichotomy theorem for the SQUARE ROOT PROBLEM is the core of a similar theorem for r^{th} ROOT PROBLEM. In almost all parts of this thesis, we are working with the square root version of the r^{th} ROOT PROBLEM.

Chapter 3

Complexity Problem of Finding Square Roots

The main motivation for studying the complexity of checking if a given graph is a certain power (square specifically) of another graph comes from *distributed computing*. Consider a network of computers that are connected through their links. We can model this network by a graph (called the underlying graph) H, where each computer is a vertex and each link is an edge between two computers, see Figure 3.1 for an example.

The r^{th} power of graph H represents the possible flow of information in r round of communication in a (distributed) network of processors (or computers) organized according to H. Linial [3] introduced a question about the characterization of this problem. He asked if there exists a polynomial time algorithm to decide the entire topology of H by knowing H^r.

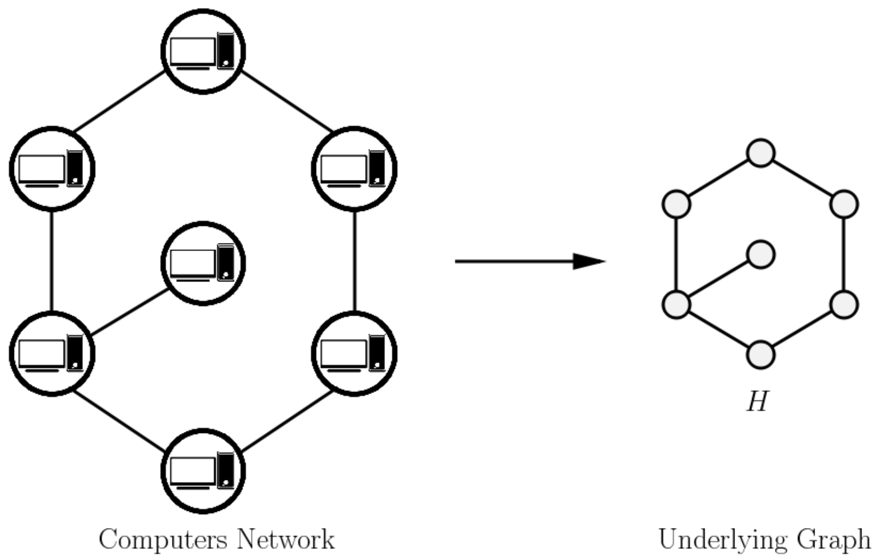

Figure 3.1: A sample of network of computers and its underlying graph.

3.1 Square Graphs and Local-Information

Let $G = H^2$. For $v \in V_H (= V_G)$ we have $N_H(v) \subseteq N_G(v)$, and $N_H(v) = N_G(v)$ only if H is a star graph. Hence we can assume that $N_G(v) - N_H(v) \neq \emptyset$. The set $N_G(v) - N_H(v)$ contains all vertices of distance 2 from v in H. Note that an algorithm that can distinguish $N_H(v)$ from $N_G(v)$ for any vertex v, can reconstruct the graph H only using information from G. Consider the square root finding problem while we know an arbitrary neighbourhood in H.

Proposition 2. *Let $G = H^2$ for some graph H with girth at least 6. Then, for all vertices $x \in V$ and all vertices $y \in N_H(x)$, $N_H(y) = N_G(y) \cap \big(N_G[x] \setminus N_H(x)\big)$.*

Proof. First, consider an arbitrary vertex $w \in N_H(y) - x$. Clearly, $w \in$

$N_G(y)$, as well $w \in N_G(x)$. Also, since H is C_3-free, $wx \notin E_H$. Thus $w \in N_G(y) \cap \big(N_G(x) \setminus N_H(x)\big)$.

Conversely, let w be an arbitrary vertex in $N_G(y) \cap \big(N_G[x] \setminus N_H(x)\big)$. Assuming $wy \notin E_H$, then $w \neq x$ and there exist vertices z and z' such that $zx, zw \in E_H$ and $z'y, z'w \in E_H$. As H is C_3-free, $zy \notin E_H$, $z'x \notin E_H$, and $zz' \notin E_H$. But then x, y, w, z and z' induce a C_5 in H, a contradiction. Thus $w \in N_H(y)$. □

Indeed if graph contains no short cycle (of length 5 or less), with knowing an arbitrary neighbourhood $N_H(v)$ we can distinguish all second neighbours and then recognize all neighbourhoods of neighbours of v. This process can be repeated to completely reconstruct the square root. In the following section we study the complexity problem for square roots without short cycles. At this point we redefine our decision problem in terms of the girth of the square root.

SQUARE OF GRAPHS WITH GIRTH g

Instance: A graph G.

Question: Does there exist a graph H of girth at least g such that $G = H^2$.

3.1.1 Square of Graphs Without Short Cycles

We study the following decision problem:

SQUARE OF GRAPHS WITH GIRTH 6 AND A SPECIFIED NEIGHBOURHOOD

Instance: A graph G, $v \in V_G$ and $U \subseteq N_G(v)$.

Question: Does there exist a graph H of girth at least six such that $G = H^2$ and $N_H(v) = U$.

Theorem 2 (Farzad et al. [11])**.** SQUARE OF GRAPHS WITH GIRTH 6 AND A SPECIFIED NEIGHBOURHOOD *has at most one solution. The unique solution, if any, can be constructed in time* $O(M(n))$[1]*.*

Proof. Given G, $v \in V_G$ and $U \subseteq N_G(v)$, assume H is a square root of G with $g(H) = 6$, such that $N_H(v) = U$. Then, by Proposition 2, the neighbourhood in H of each vertex $u \in U$ is uniquely determined by $N_H(u) = N_G(u) \cap \big(N_G[v] \setminus U\big)$. By repeatedly applying Proposition 2 for each $u \in U$ and $U' = N_H(u)$, we can conclude that H is unique. Assuming the pseudo-code given in appendix A, It can be seen, that H is a graph of girth at least 6, correctness of pseudo-code follows from Proposition 2. Since every vertex is enqueued at most once, it (lines 1-13 in pseudo code) take $O(m)$ steps, $m = |E_G|$. Checking if $G = H^2$ (line 14 in pseudo code) takes $O(M(n))$ steps, $n = |V_G|$. □

Although knowing an arbitrary neighbourhood is not changing the complexity nature of the problem, but we can find a square root with a rather stricter condition.

Proposition 3. *Suppose H is of girth at least 6, $xy \in E_H$ and $H_2 = G$. Then $C = N_G(x) \cap N_G(y)$ has at most two connected components. Moreover,*

[1]Note that $M(n)$ stands for the time needed to perform a matrix multiplication of two $n \times n$ matrices; currently, $M(n) = O(n^{2.376})$.

if A and B are the connected components of C (one of them maybe empty) then

- $A = N_H(y) - \{x\}, B = N_H(x) - \{y\}$

- $A = N_H(x) - \{y\}, B = N_H(y) - \{x\}$

Proof. One can see C has at most two connected components. Since $xy \in E_H$ vertices that are adjacent to y are either neighbours of x or y (in G) hence we have $A = N_H(y) - \{x\}, B = N_H(x) - \{y\}$ or $A = N_H(x) - \{y\}, B = N_H(y) - \{x\}$. □

The stricter version of the problem is as follows:

Square of Graphs With Girth 6 and One Specified Edge

Instance:	A graph G, $v \in V_G$ and and edge $xy \in E_G$.
Question:	Does there exist a graph H of girth at least six such that $G = H^2$ and $xy \in E_H$.

Theorem 3 (Farzad et al. [11]). Square of Graphs with Girth 6 and One Specified Edge *can be solved in time $O(n^4)$.*

It has been proven by using Proposition 3.

Theorem 4 (Farzad et al. [11]). *Square of Graphs with Girth 6 can be solved in time $O(\delta(G)n^4)$.*

It has been proven by using Theorem 3.

3.1.2 Square of Graphs With Short Cycles

Proposition 2 is the heart of the known efficient algorithms in the study of square root finding problem. As one can see the condition "girth at least six" is necessary for Proposition 2. In fact we will see in the following two subsections and Chapter 4 that there is no efficient algorithm to decide if there exists a square root when the girth of square root graph is less than five.

When the complexity of square root finding was introduced by Linial [3] in 1992 for the first time, Motwani and Sudan [23] answered it fairly quickly in 1994. They proved it is NP-complete to determine if a given graph is the square of another graph. Indeed they proved the SQUARE ROOT problem is NP-complete. They reduce the NAESAT[2] problem (which is an special version of 3-SAT problem) to the SQUARE ROOT problem. Although they proved SQUARE ROOT is NP-complete, the square root graph which they used in their reduction contains many triangles and is relatively dense.

A main source of interest for solving this problem comes from distributed networks. Distributed graphs are rarely dense and are often of a certain girth. Hence we are more interested to find a complete dichotomy theorem for this problem in terms of the girth of the square root. It is also worth while mentioning that the reduction introduced by Motwani and Sudan [23] can not be extended for any other girth of the square root.

In the following two sections we introduce reductions of [23] and [11],

[2] *Not-All-Equal 3SAT*

however we just present the sketch of their results.

Square of Graphs With Girth Three

Theorem 5. *(Motwani and Sudan [23])* Square of Graphs With Girth Three *is NP-complete.*

Motwani and Sudan [23] reduce the NAESAT to Square of Graphs With Girth Three. NAESAT is like 3SAT, but we require additionally that there be a satisfying truth assignment under which no clauses have the three literals equal in truth value. In other words each clause must have one or two literal assigned true.

NAESAT is an NP-complete problem [31]. Although we do not deal with details of their proof, we present the sketch of their proof.

Basic Theme of The Reduction Graph: We are about to introduce a (reduction) square graph $G(\phi)$ where ϕ is an instance of NAESAT. Before defining $G(\phi)$ specifically, lets give a general idea about it. A proper construction must present a square graph that does not contain information of any assignment but its square root does. We choose two constant vertices $TRUE$ and $FALSE$ and represent each literal l_i as a vertex called L_i. L_i is adjacent to both $TRUE$ and $FALSE$ in $G(\phi)$ (as it contains no information of any assignment) but to exactly one of $TRUE$ or $FALSE$ in square root of $G(\phi)$. The graph shown in Figure 3.1.2 is a basic illustration of this idea for a clause c_j containing three literals x, y and z.

Vertices A_{xy}, A_{yz} and A_{xz} are added to make sure all vertices are adjacent

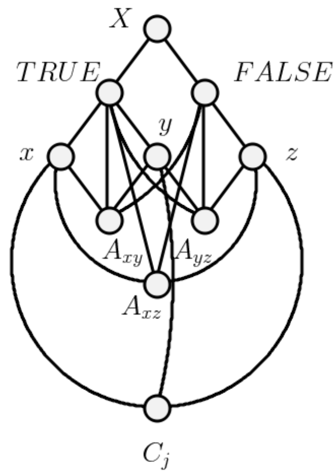

Figure 3.2: A subgraph of $H = \sqrt{G(\phi)}$ for representing a clause

to $TRUE$ and $FALSE$ in $G(\phi)$. Also vertex C_j that represents c_j, is also to make sure all x, y and z do not have the same truth value (otherwise the vertex C_j is not adjacent to both $TRUE$ and $FALSE$ in $G(\phi)$).

While literal vertices have different possibilities in connecting to constant vertices $TRUE$ and $FALSE$, we need to make sure that $X \sim \{TRUE, FALSE\}$ and $C_j \sim \{x, y, z\}$. The idea introduced by Motwani and Sudan is a construction called *tail*. Tail is a P_4 (a path with four vertices) associated with the desired (fixed) neighbourhood. For example if we associate the vertex X with onw end of a P_4 in the square root graph, then we can make sure that $X \sim \{TRUE, FALSE\}$ in any square root of $G(\phi)$, see Figure 3.1.2. Same idea works for C_j as well.

Reduction Let c_j be the set of literals in clause j and let $\mathcal{C} = \{c_j | 1 \leq$

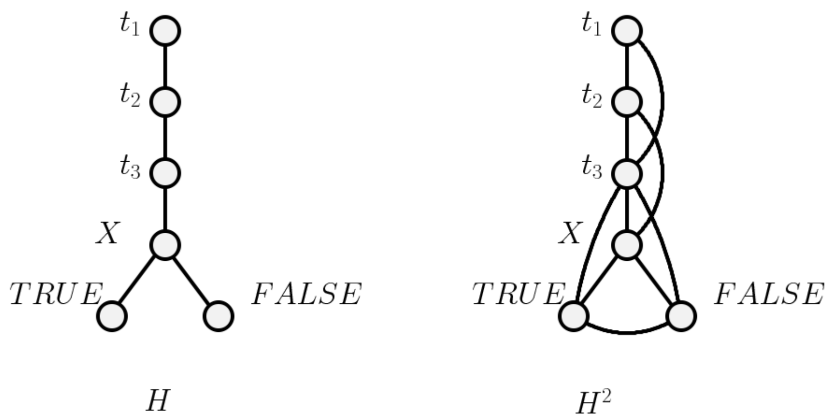

Figure 3.3: The *tail* structure.

$j \leq m\}$. The graph $G(\phi)$ is constructed as follows:

- $V_{G(\phi)}$ contains:
  - Constant Vertices: $TRUE$, $FALSE$, X and tail vertices to X, t_1, t_2 and t_3.
  - Literal Vertices: $L_i : 1 \leq i \leq 2n$ for each literal l_i.
  - Literal Pair Vertices: $A_{ij} : 1 \leq i < j \leq 2n$ for each literal $l_i \neq \neg l_j$.
  - Clause Vertices: C_j for each clause $c_j \in \mathcal{C}$ and tail vertices C_j^1, C_j^2 and C_j^3.
- $E_{G(\phi)}$ contains:
  - Edges of the tail of X: $t_3 \sim t_2$, $t_3 \sim t_1$, $t_2 \sim t_1$, $t_2 \sim X$, $t_1 \sim \{X, TRUE, FALSE\}$, $X \sim \{TRUE, FALSE\}$ and $TRUE \sim$

$FALSE$.

- Edges of the tail of clauses: $\forall c_j \in \mathcal{C}$, $C_j^3 \sim C_j^2$, $C_j^3 \sim C_j^1$, $C_j^2 \sim C_j^1$, $C_j^2 \sim C_j$, $C_j^1 \sim \{L_{i1}, L_{i2}, L_{i3}\}$ where $c_j = \{l_{i1}, l_{i2}, l_{i3}\}$.
- Edges of the clauses: $\forall c_j \in \mathcal{C}$, $\forall i$ such that $l_i \in c_j$, $C_j \sim \{L_i, TRUE, FALSE\}$, also $\forall k$ such that $l_k \neq \neg l_i$, $C_j \sim A_{ik}$.
- Edges of the literal vertices: All remaining vertices $\{L_i\}$'s, the $\{A_{ij}\}$'s and $TRUE, FALSE$ and X are adjacent, except for the the pairs L_i and L_j where $l_i = \neg l_j$.

One can see that this graph can be constructed for a given ϕ in polynomial time. To prove Square of Graphs With Girth Three is NP-complete we just need to show that for ϕ (an instance of NAESAT), when there is an assignment in such a way that ϕ evaluates as true then $G(\phi)$ is the square of some graph. The graph $H = G(\phi)^2$ is constructed as follows:

$E_{G(\phi)}$ contains:

- Edges of the tail of X: $t_3 \sim t_2$, $t_2 \sim t_1$, $X \sim \{t_1, TRUE, FALSE\}$.
- Edges of the tail of clauses: $\forall c_j \in \mathcal{C}$, $C_j^3 \sim C_j^2$, $C_j^2 \sim C_j^1$, $C_j^1 \sim \{C_j^1, L_{i1}, L_{i2}, L_{i3}\}$ where $c_j = \{l_{i1}, l_{i2}, l_{i3}\}$.
- Edges of the literal vertices: according to the assignment if l_i evaluated as true then L_i is adjacent to $TRUE$ and to $FALSE$ otherwise. Also for each literal l_k that $l_k \neq \neg l_i$, $L_i \sim A_{ik}$.

- Edges of literal pair vertices A_{ik}: $A_{ik} \sim TRUE, FALSE$.

One can see that $H^2 = G(\phi)$. And also a square root of $G(\phi)$ can be used to form a truth assignment for ϕ.

Square of Graphs With Girth Four

Farzad et al. [11] proved that the problem of The Square of Graphs With Girth Four is NP-complete. They reduced the set splitting problem to The Square of Graphs With Girth Four. The nature of the set splitting problem is reasonably simple, and construction of the reduction graph does not need any preliminary explanation.

Theorem 6 (Farzad et al. [11])**.** Square Root Graph with Girth Four *is NP-complete.*

Sketch of the proof: They reduce the following NP-complete problem set splitting [12, Problem SP4], also known as hypergraph 2-colorability, to it.

set splitting

Instance:	Collection D of subsets of a finite set S.
Question:	Is there a partition of S into two disjoint subsets S_1 and S_2 such that each subset in D intersects both S_1 and S_2?

Let $S = \{u_1, \ldots, u_n\}$, $D = \{d_1, \ldots, d_m\}$ where $d_j \subseteq S$, $1 \leq j \leq m$, be an instance of set splitting. We construct an instance $G = G(D, S)$ for The Square of Graphs With Girth Four as follows.

The vertex set of graph G consists of:

- U_i, $1 \leq i \leq n$. Each 'element vertex' U_i corresponds to the element u_i in S.

- D_j, $1 \leq j \leq m$. Each 'subset vertex' D_j corresponds to the subset d_j in D.

- D_j^1, D_j^2, D_j^3, $1 \leq j \leq m$. Each three 'tail vertices' D_j^1, D_j^2, D_j^3 of the subset vertex D_j correspond to the subset d_j in D.

- S_1, S_1', S_2, S_2', four 'partition vertices'.

- X, a 'connection vertex'.

The edge set of graph G consists of:

- Edges of tail vertices of subset vertices:

  For all $1 \leq j \leq m$: $D_j^3 \sim D_j^2$, $D_j^3 \sim D_j^1$, $D_j^2 \sim D_j^1$, $D_j^2 \sim D_j$, $D_j^1 \sim D_j$, and for all i, $D_j^1 \sim U_i$ whenever $u_i \in d_j$.

- Edges of subset vertices:

  For all $1 \leq j \leq m$: $D_j \sim S_1$, $D_j \sim S_1'$, $D_j \sim S_2$, $D_j \sim S_2'$, $D_j \sim X$, $D_j \sim U_i$ for all i, and $D_j \sim D_k$ for all k with $d_j \cap d_k \neq \emptyset$.

- Edges of element vertices:

  For all $1 \leq i \leq n$: $U_i \sim X$, $U_i \sim S_1$, $U_i \sim S_2$, $U_i \sim S_1'$, $U_i \sim S_2'$, and $U_i \sim U_{i'}$ for all $i' \neq i$.

- Edges of partition vertices:

  $S_1 \sim X$, $S_1 \sim S_1'$, $S_1 \sim S_2'$, $S_2 \sim X$, $S_2 \sim S_1'$, $S_2 \sim S_2'$, $S_1' \sim X$, $S_2' \sim X$.

Clearly, G can be constructed from D, S in polynomial time. Now if there exists a partition of S into two disjoint subsets S_1 and S_2 such that each subset in D intersects both S_1 and S_2, then there exists a graph H with girth four such that $G = H^2$.

Since depicting H, the square root of $G(S, D)$ depends on the answer to set splitting problem we illustrate an example of H for $S = \{u_1, u_2, \ldots, u_5\}$ and $D = \{d_1, d_2, d_3, d_4\}$ with $d_1 = \{u_1, u_2, u_3\}$, $d_2 = \{u_2, u_5\}$, $d_3 = \{u_3, u4\}$, and $d_4 = \{u_1, u_4\}$. The graph shown in Figure 3.4 is $G(S, D)$.

The graph shown in Figure 3.5 is H, the square root of $G(S, D)$.

This reduction shows that SQUARE ROOT GRAPH WITH GIRTH FOUR is NP-complete.

3.2 Uniqueness of Square Roots

In Subsection 3.1.1 we introduced an efficient algorithm that computes a square root of girth at least six of a given graph. However as we showed in Subsection 3.1.2, there is no efficient algorithm to compute a square root of girth three or four (if it exists). In this section we present the results about the uniqueness (up to isomorphism) of square roots.

The uniqueness of the square root is important from the complexity point

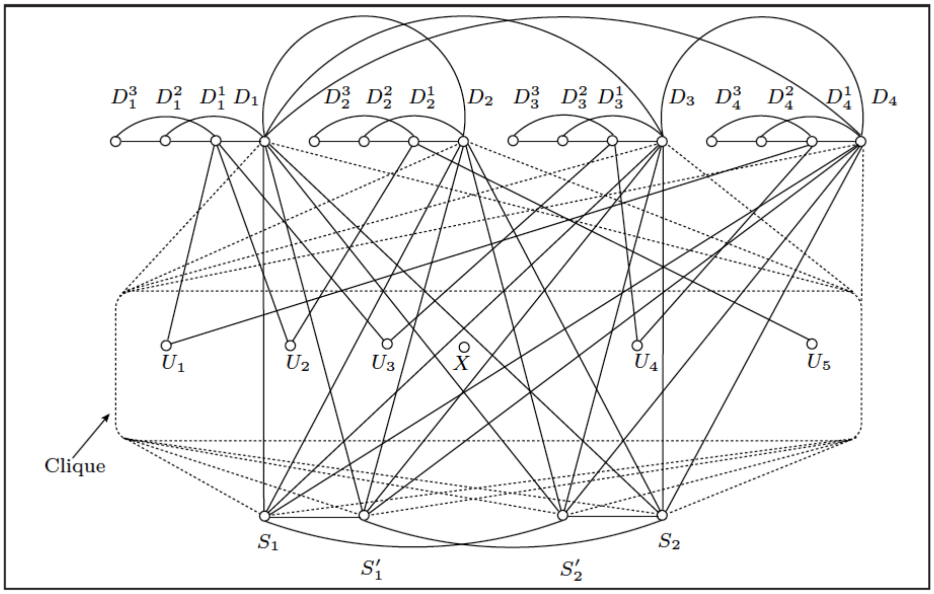

Figure 3.4: Reduction graph for Square of Graphs With Girth Four, Figure 2 from [11]

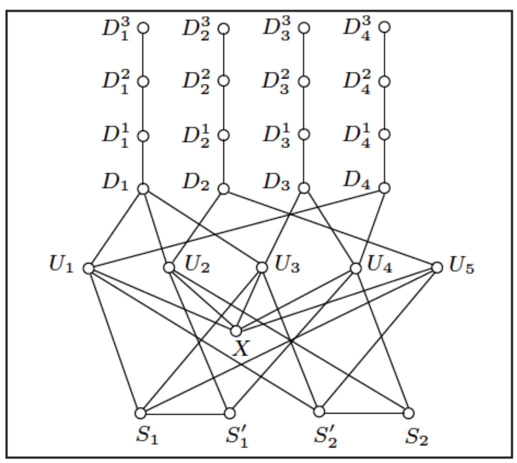

Figure 3.5: The Square Root Graph H, $H^2 = G(S, D)$, Figure 3 from [11]

of view. Except the unknown case square root graphs of girth 5, when the *Square Root of Girth g* is polynomial then the output of an efficient algorithm is unique and when the problem is NP-complete, there are graphs of girth g with more than one square root. We will see how this fact turns to a sign in the recognition problem of square graphs with girth five, in Chapter 4.

Farzad et al. [11] is providing the uniqueness of square root with girth at least 7.

Definition 4. *Let G be an arbitrary graph. An edge of G is called forced if it is contained in (at least) two distinct maximal cliques in G.*

Theorem 7 (Farzad et al. 2009)**.** *The square roots with girth at least seven of squares of graphs with girth at least seven are unique, up to isomorphism.*

Proof. Let G be the square of some graph H with girth ≥ 7. If G is complete, clearly, every square root with girth ≥ 6 of G must be isomorphic to the star $K_{1,n-1}$ where n is the vertex number of G.

Thus, let G be non-complete, and let F be the subgraph of G formed by the forced edges. If F has only one edge, G clearly consists of exactly two maximal cliques, Q_1, Q_2, say, and $Q_1 \cap Q_2$ is the only forced edge of G. Then, it is easily seen that every square root with girth ≥ 6 of G must be isomorphic to the double star T having center edge v_1v_2 and $\deg_T(v_i) = |Q_i|$.

So, assume F has at least two edges. Then for each two maximal cliques Q, Q' in G with $Q \cap Q' = \{x, y\}$, x or y is the unique center vertex of the star $F[V_F \cap Q]$ or $F[V_F \cap Q']$. Hence, for any end-vertex u of H, i.e., $u \in V_G - V_F$, the neighbor of u in F is unique. Since F is the graph resulting from H by deleting all end-vertices, H is therefore unique. □

Adamaszek and Adamaszek [1] proved that the square root problem proved by Farzad et al. [11] also give us the only possible square root.

Theorem 8 (Anna Adamaszek, Michal Adamaszek [1]). *If H_1 and H_2 are two graphs of girth at least 6 such that $H_1^2 \cong H_2^2$ then $H_1 \cong H_2$.*

Proof. We first prove that if there exists a xyz-path in both H_1 and H_2 then $H_1 \cong H_2$. Let $G = H^2$, for some H of girth six, then we have, (see Figure 3.6):

$$N_H(x) = (N_G(x) \cap N_G(y))\ N_G(z) \cup \{x, y\}$$

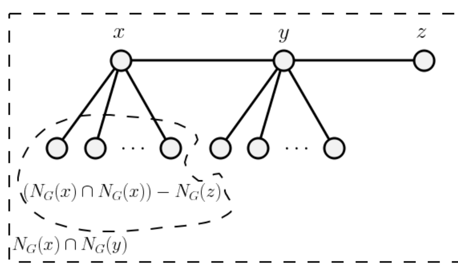

Figure 3.6: $N_H(x) = (N_G(x) \cap N_G(y))\ N_G(z) \cup \{x, y\}$ for an xyz-path in H, where $g(H) \geq 6$.

and if $d_H(y) = 1$, i.e. there exist no z, then:

$$N_H(y) = N_G(x)$$

With the above formulas, given one path xyz of H one can recursively compute all the edges of H using only the information from G, so the square root of G with this distinguished path is unique.

So we may assume for every v the set $X_v = \{u \dim uv \in E(H_1) \cap E(H_2)\}$ has at most 1 element. Define the following map: $f : V \to V$

- if $|X_v| = 0$ then $f(v) = v$

- if $|X_v| = 1$ then $F(v)$ is the unique element of X_v.

which is an isomorphism from H_1 to H_2. □

3.3 Square Root Problem So Far

Putting all results together Table 3.1 represents the square root problem and fastest known algorithms regarding the girth of square root graph (H).

Girth	Complexity Class	Uniqueness of Root
∞	$O(\|V\| + \|E\|)$ [7]	Up to labelling
$g(H) \geq 7$	$O(\|V\| \times \|E\|)$ [11]	Up to labelling [11]
$g(H) \geq 6$	$O(\|V\| \times \|E\|)$ [11]	Up to isomorphism [1]
$g(H) \geq 5$	?	No
$g(H) \geq 4$	NP-Complete [11]	No
$g(H) \geq 3$	NP-Complete [23]	No

Table 3.1: Known complexity classes for Square of Graphs With Girth g.

The recognition problem has been open for square roots of girth 5. In Chapter 4 we show that this problem is NP-complete. The result is providing a complete dichotomy complexity theorem for Square of Graphs With Girth g. Before we introduce our reduction, we present a family of graphs with exponentially many non-isomorphic square roots. This family was indeed the sign that leads us through the NP-completeness of the problem.

The interest in the complexity problem of SQUARE OF GRAPHS WITH GIRTH 5 is not limited to this dichotomy theorem. There are other open problems related to SQUARE OF GRAPHS WITH GIRTH 5.

The complete graph $G = K_n$ has a square root in the graphs with finite girth if and only if there exists a graph on n vertices that has girth 5 and diameter 2, a Moore graph. By the Hoffman-Singleton theorem (see [28]) such a graph may exist only for $n = 5, 10$ and 50. See Figure 3.7 for a Moore graph of girth five diameter 2 and 50 vertices.

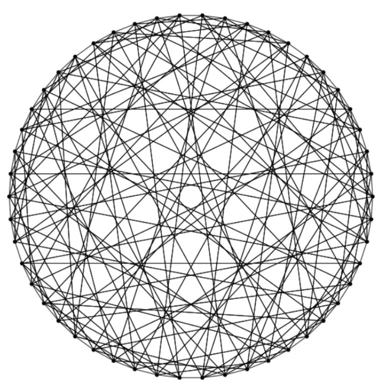

Figure 3.7: Moore Graph of Girth 5 and Diameter 2 on 50 vertices.

The existence of such a graph on 3250 vertices is a long time open problem. Therefore, any efficient algorithm for the square root problem might (at least in principle) solve this problem.

Chapter 4

Square of Graphs With Girth Five

As we saw in Chapter 3 the recognition problem of Square of Graphs With Girth g has an internal connection with the uniqueness of square root of girth g. Indeed excluding the unknown case of girth 5, when Square of Graphs With Girth g has an efficient algorithm to recognize a square root graph, the square root is unique up to isomorphism. Also when Square of Graphs With Girth g is NP-complete then there are graphs with many non-isomorphic square roots of girth g.

In Section 4.1, we advance the technique used in the auxiliary problem of Square of Graphs With Girth at Least 6 and A Specified Neighbourhood for graphs with girth 5. This leads us to a polynomial time algorithm for square of graphs without dense subgraphs.

In Section 4.2, we study the problem of Square of Graphs With Girth 5 with a question about the uniqueness of square root. We construct a family of graphs with exponentially many non-isomorphic square roots.

We disproved the conjecture in [11] in Section 4.3, by showing that SQUARE OF GRAPHS WITH GIRTH 5 is NP-complete. This theorem is presenting the complete dichotomy theorem for the girth-parametrized square root finding problem.

4.1 Square Root Graphs Without Dense Subgraphs

In this section we present a graph called H_1, such that when H_1 is excluded from the square root, the square root finding problem can be solved in polynomial time. A stronger result is provided in Subsection 4.1.2 by excluding two edge intersecting H_1s from H.

4.1.1 Characterization of Square of Graphs With no H_1

Definition 5. *Let $G = H^2$, for $v \in V_G$ let $x \in N_G(v) - N_H(v)$, we define $L_v(x)$ as follows:*

$$L_v(x) = \{u \in N_H(v) \mid ux \in E_G\} = N_G(x) \cap N_H(v)$$

Let $G = H^2$ such that $g(H) = 5$ and H contains no H_1 as a subgraph (see Figure 4.1).

We define $\mathcal{H}_1$ to be the family of all graphs of girth at least five with no subgraph of H_1.

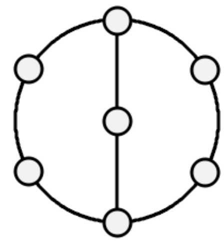

Figure 4.1: H_1

Lemma 1. *Let $G = H^2$ where $H \in \mathcal{H}_1$. For $v \in V$ and $x \in N_G(v) - N_H(v)$*

(i) $1 \leq |L_v(x)| \leq 2$

(ii) If $L_v(x) = \{u\}$ then $x \in N_H(u)$

Proof. (i) It is trivial that $L_v(x)$ has at least one element, and considering a $C_5 = [x, u, x, x', u']$ we have $L_v(x) = \{u, u'\}$. We want to show that there exists no x such that $L_v(x) = \{u, u', u''\}$. Assume otherwise, we know that x is in the neighbourhood of one of u, u' or u'', otherwise it is not in $N_G(v)$ or $L_v(x)$ has more than three vertices. Without loss of generality let $x \in N_H(u)$, therefore there exist two different vertices x' and x'' such that $xx'u', xx''u'' \subseteq E_H$, see Figure 4.2, which is a contradiction with non-existence of H_1 in H.

(ii) Since $x \in N_G(v) - N_H(v)$ then $ux \in E_H$ which means $x \in N_H(u)$.

□

Lemma 2. *Let $G = H^2$ such that $H \in \mathcal{H}_1$. For $v \in V$ and $x \in N_G(v) -$*

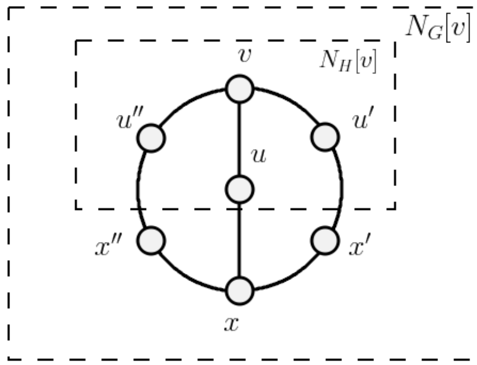

Figure 4.2: Lemma 1-i

$N_H(v)$, let $L_v(x) = \{u, u'\}$. For $y \in N_G(v) - N_H(v)$ such that $L_v(y) = \{u\}$ if $xy \in E_G$ then $x \in N_H(u)$ otherwise $x \in N_H(u')$.

Proof. Assume otherwise and let $x \in N_H(u')$ while $xy \in E_G$. It means there exists a vertex x' such that $xx'u \subseteq E_H$. Since $L_v(y) = \{u\}$ then $y \in N_H(u)$ (by lemma 1-i), therefore $yx \notin E_H$ otherwise we have a cycle of length four in H. So there exists y' such that $yy'x \subseteq E_H$, see Figure 4.3. And this is a contradiction due to non-existence of H_1 in H. Therefore $x \in N_H(u)$. Since every neighbour of u is connected to y then if $xy \notin E_G$ then $x \notin N_H(u)$ which means $u \in N_H(u')$. $\square$

Knowing the neighbourhood of (an arbitrary vertex) v in H, for every $x \in N_G(v) - N_H(v)$, Lemma 1 says we have two possibilities, either $L_v(x) = \{u\}$ which means $vu \in E_H$, or $L_v(x) = \{u, u'\}$. In the second case we need Lemma 2 to decide $xu \in E_H$ or $xu' \in E_H$, and this gives us a procedure to decide the neighbourhood of each $u \in N_H(v)$. Using $N_H(v) = U$ we can build

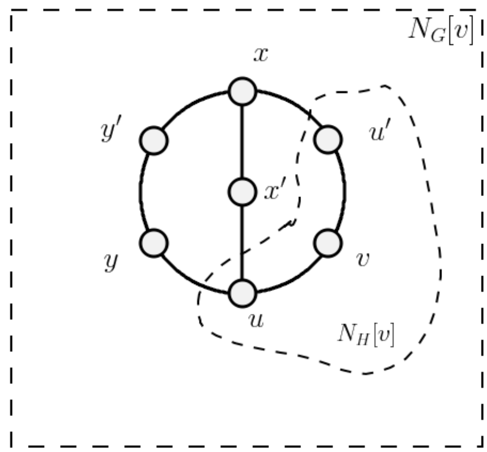

Figure 4.3: Lemma 2

a BFS-like algorithm to decide the whole H. A sample of such pseudo-code is in *Appendices-B*.

4.1.2 Characterization of Square of Graphs With no Edge Intersecting H_1s

We can advance further by including subgraphs isomorphic to H_1, but excluding two edge intersecting H_1s.

We define $\mathcal{H}_2$ to be a family of all graphs of girth at least five with no subgraph of two edge intersecting H_1s.

Lemma 3. *Let $G = H^2$ where $H \in \mathcal{H}_2$. For $v \in V$ and $x \in N_G(v) - N_H(v)$:*

(i) $1 \leq |L_v(x)| \leq 3$

(ii) If $L_v(x) = \{u\}$ *then* $x \in N_H(u)$

Proof. (i) It is trivial that $L_v(x)$ has at least one element, and considering a $C_5 = [x, u, x, x', u']$ we have $L_v(x) = \{u, u'\}$. Considering an H_1 also gives us a vertex x such that $L_v(x) = \{u, u', u'\prime\}$. We want to show that there exists no x such that $L_v(x) = \{u_1, u_2, u_3, u_4\}$. Assume otherwise, we know that x is in neighbourhood of one of u_1, u_2, u_3 or u_4, otherwise it is not in $N_G(v)$ or $L_v(x)$ has more than four vertices. Without loss of generality let $x \in N_H(u_1)$, therefore there exist three different vertices x_2, x_3 and x_4 such that $xx_2u_2, xx_3u_3, xx_4u_4 \subseteq E_H$, see Figure 4.4, which is a contradiction with existence of two intersecting H_1.

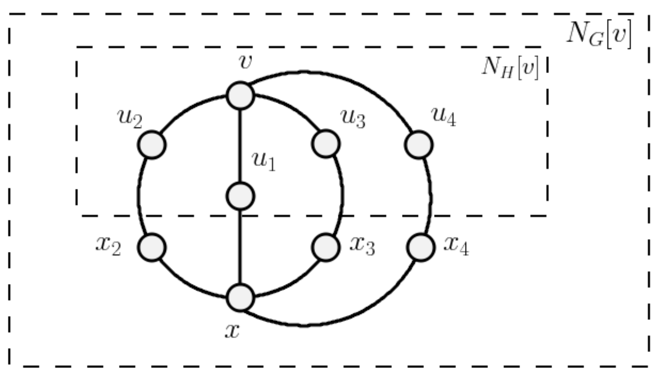

Figure 4.4: Lemma 3-*i*

(ii) Since $x \in N_G(v) - N_H(v)$ then $ux \in E_H$ which means $x \in N_H(u)$.

□

Lemma 4. *Let $G = H^2$ such that $H \in \mathcal{H}_2$. For $v \in V$ and $x, y \in N_G(v) - N_H(v)$, if $|L_v(x) \cap L_v(y)| \leq 1$ then $xy \notin E_H$*

Proof. Assume otherwise and let $xy \in E_H$, then there exists $u, w \in N_H(v)$ such that $ux, wy \in E_H$ which means $\{u, w\} \subseteq L_v(x) \cap L_v(y)$ and is a contradiction to $|L_v(x) \cap L_v(y)| \leq 1$. □

Lemma 5. *Let $G = H^2$ such that $H \in \mathcal{H}_2$. For $v \in V$ and $x \in N_G(v) - N_H(v)$, let $L_v(x) = \{u_1, u_2, u_3\}$. Also let $S_x = \{y \in (N_G(v) - N_H(v)) \cap N_G(x) \mid L_v(x) \cap L_v(y)| \geq 2\}$. Then we have $xu_i \in E_H$ for $\{u_i\} = \bigcap_{y \in S} L_v(y)$.*

Proof. Since $x \in N_G(v) - N_H(v)$, $xu_j \in E_H$ for some $u_j \in L_v(x)$. Without loss of generality let $xu_1 \in E_H$. It is enough to show that $\{u_1\} = \bigcap_{y \in S} L_v(y)$.

- $u_1 \in \bigcap_{y \in S_x} L_v(y)$: Assume otherwise and let $y \in S$ and $xy \notin E_H$ otherwise $u_1 \in L_v(y)$. Since $|L_v(x) \cap L_v(y)| \geq 2$ and $u_1 \notin L_v(y)$ then $L_v(x) \cap L_v(y) = \{u_2, u_3\}$. Having $yu_2 \in E_H$ or $yu_3 \in E_H$ is a contradiction with having two intersecting H_1 in H. So there exists y_2, y_3 such that $yy_2u_2, yy_3u_3 \subseteq E_H$. Also let $w \in N_H(v)$ such that $w \notin L_v(x)$ and $yw \in E_H$. Now $[w, v, y, u_2, u_3, y_2, y_3]$ and $[u_1, v, x, u_2, u_3, x_2, x_3]$ are two different intersecting H_1.

- $\bigcap_{y \in S} L_v(y) \subseteq \{u_1\}$, it is trivial that having u_i and u_j in $\bigcap_{y \in S} L_v(y)$ forming two intersecting H_1.

□

Knowing the neighbourhood of (an arbitrary vertex) v, for every $x \in N_G(v) - N_H(v)$ Lemma 3 says we have three possibilities, either $L_v(x) = \{u\}$ which means $vu \in E_H$ or $L_v(x) = \{u_1, u_2, u_3\}$ or $|L_v(x)| = 2$. In the second case we need Lemma 5 to decide for which u_i, $xu_i \in E_H$. Furthermore it gives us the parent for other vertices connecting to x that and having L_v of size two. In third case, we have no other vertices adjacent to x having L_v of size three and Lemma 2 is again useful in this case. Again we can come up with a procedure to decide the neighbourhood of each $u \in N_H(v)$. Using $N_H(v) = U$ we can build a BFS-like (the following) algorithm to decide the whole H. A sample of such pseudo-code is in *Appendices-C*.

4.2 Graphs with Many Non-Isomorphic Square Roots of Girth Five

For a given graph G if there exists H where $G = H^2$ and $g(H) \geq 6$, then H is unique up to isomorphism [1]. However this is not true when the girth of H is at least 5. For $G = K_5$, two graphs $K_{1,4}$ and C_5 are non-isomorphic square roots of G. These two graphs can be used to introduce a family of non-isomorphic pairs of graphs with the same square, see Figure 4.5.

Notice that graphs in this family contain vertices of degree 1. Such vertices were a main source of technicalities in the past studies.

In [1] there is also an example of a graph with two non-isomorphic square root of girth five, see Figure 4.6. These two graphs are more interesting as, unlike the graphs shown in Figure 4.5, they contain no vertex of degree 1. These two graphs are also the smallest non-isomorphic graphs with girth five, minimum degree 2 and identical squares. In this thesis, we call these two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$.

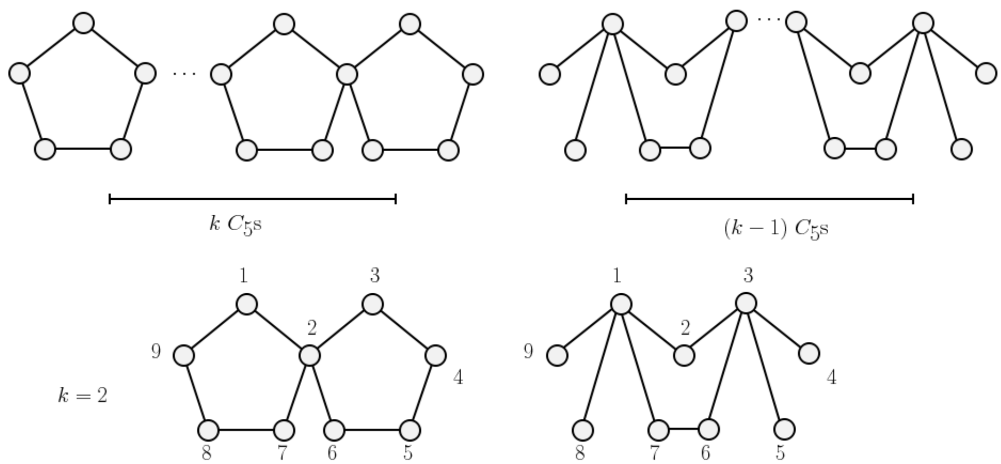

Figure 4.5: A family of non-isomorphic graphs with identical square graph.

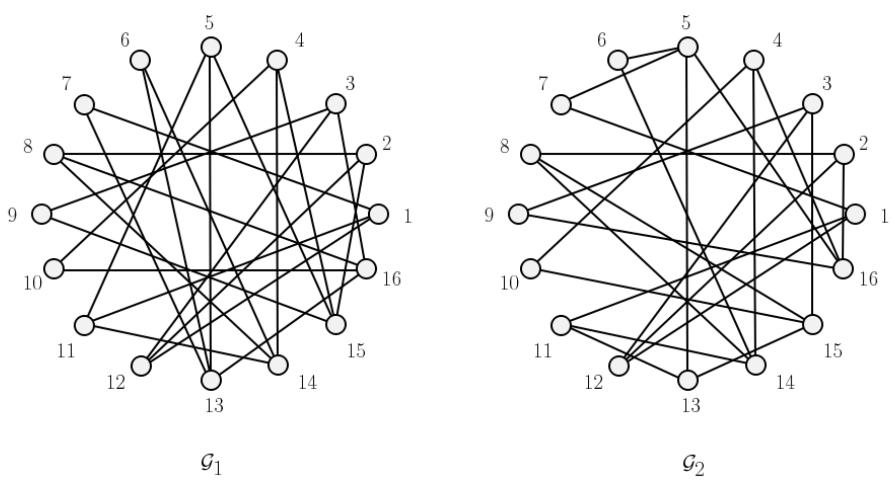

Figure 4.6: Non-isomorphic graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ with no vertex of degree 1 and identical squares.

It is also an interesting question to ask if there exists a graph with many non-isomorphic square roots. We show that $\mathcal{G}_1$ and $\mathcal{G}_2$ can be used to construct a family of graphs with many non-isomorphic square roots. With current labelling of $\mathcal{G}_1$ and $\mathcal{G}_2$, we have three vertices $1, 12$ and 14, that their neighbourhoods in both $\mathcal{G}_1$ and $\mathcal{G}_2$ are identical. So we may identify two graphs on one of these three vertices to construct a new graph with more than one square roots. For example, we can identify vertex 1 in both $\mathcal{G}_1$ and $\mathcal{G}_2$ as shown in Figure 4.7.

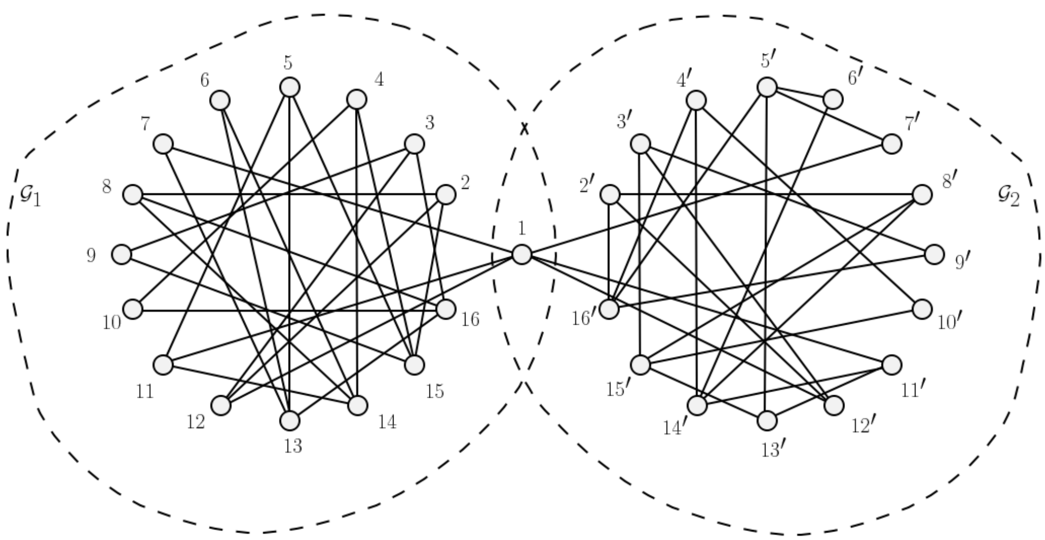

Figure 4.7: Connecting $\mathcal{G}_1$ and $\mathcal{G}_2$ by identifying vertex 1.

Observation 1. *The square of the graph shown in Figure 4.7 has three non-isomorphic square roots.*

Proof. In Figure 4.7, by replacing the copy of $\mathcal{G}_1$ on the vertices $\{1, 2, \ldots, 16\}$

with a copy of $\mathcal{G}_2$, we would get a different graph with the same square. Hence switching copies of $\mathcal{G}_1$ and $\mathcal{G}_2$ constructs three non-isomorphic graphs with identical squares. □

The process of connecting $\mathcal{G}_1$s and $\mathcal{G}_2$s by identifying one of those three vertices can form a family of graphs with girth five, minimum degree of 2 and exponentially many non-isomorphic square roots. See Figure 4.8 for an illustration of 16 non-isomorphic graphs with identical square.

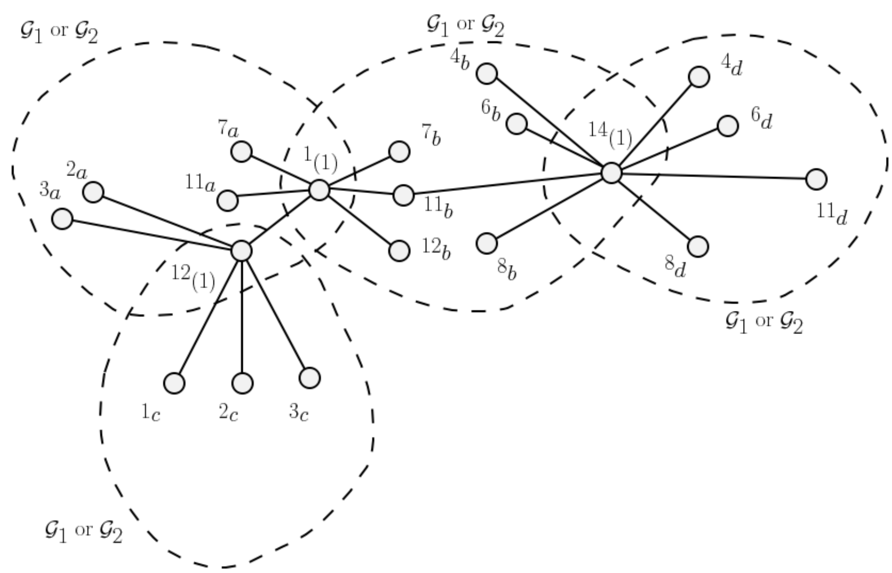

Figure 4.8: Non-isomorphic graphs with identical square.

This process is introducing a family of graphs with exponentially many non-isomorphic square roots. This family indicates that even with the restriction $\delta_H \geq 2$ knowledge of any local neighbourhood is not sufficient to reconstruct the rest of the square root.

When a square root has no short cycle (girth of at least 6) the square root finding problem is solvable by an efficient algorithm [11]. The main idea of this algorithm (and almost all attempts to find an efficient algorithm for the square root finding problem) is to use a known neighbourhood of the square root and then reconstruct the whole square root by only using informations from the square graph. Indeed if we know an arbitrary neighbourhood of graph H of girth at least six, where $H^2 = G$, then we can recognize second neighbours (vertices of distance two) of that vertex. In this way the whole graph H can be uniquely reconstructed only using information of G. The family of graphs we introduced using $\mathcal{G}_1$ and $\mathcal{G}_2$ indicates that by knowing an arbitrary neighbourhood of the square root we can never decide the rest of the graph, as there are always options (to decide a second neighbourhood of a vertex) that result in different (non-isomorphic) graphs. Hence knowing a constant number of neighbourhoods in the square root can not help to find a square root for a given graph (or to decide if there exists a square root).

We also use $\mathcal{G}_1$ and $\mathcal{G}_2$ graphs as part of our reduction in Section 4.3. We need to show that the graph $\mathfrak{G} = {\mathcal{G}_1}^2 = {\mathcal{G}_2}^2$ has only two non-isomorphic square roots which are $\mathcal{G}_1$ and $\mathcal{G}_2$. For the rest of this chapter we use $\mathfrak{G}$ as the square of $\mathcal{G}_1$ (or $\mathcal{G}_2$).

Lemma 6. *Let $\mathfrak{G} = H^2$ for $g(H) = 5$, then $N_H(1) = \{7, 11, 12\}$.*

Proof. We show this in the following four steps:

I $1 \overset{H}{\sim} 12$: Assume otherwise and let $1 \overset{H}{\nsim} 12$, now since $1 \overset{\mathfrak{G}}{\sim} 12$ then

$N_H(1) \subseteq (N_{\mathfrak{G}}(12) \cap N_{\mathfrak{G}}(1)) = \{11, 7, 2, 3\}$. In other hand we have $11 \overset{\mathfrak{G}}{\sim} 7$, $2 \overset{\mathfrak{G}}{\sim} 3$ but non of 11 or 7 is not adjacent to any of 2 and 3, therefore either $N_H(1) \subseteq \{2, 3\}$ or $N_H(1) \subseteq \{7, 11\}$. If $N_H(1) \subseteq \{2, 3\}$ then we have a contradiction with $11 \in N_{\mathfrak{G}}(1)$, and if $N_H(1) \subseteq \{7, 11\}$ we again have a contradiction with $2 \in N_{\mathfrak{G}}(1)$, this implies $1 \overset{H}{\sim} 12$.

II $1 \overset{H}{\sim} 11$: Assume otherwise and let $1 \overset{H}{\nsim} 11$, now since $1 \overset{\mathfrak{G}}{\sim} 11$ then $N_H(1) \subseteq (N_{\mathfrak{G}}(11) \cap N_{\mathfrak{G}}(1)) = \{7, 5, 12, 13, 14\}$. But according to part I, we know that $1 \overset{H}{\sim} 12$, therefore $N_H(1) \subseteq \{7, 12\}$, and this is a contradiction because non of 12 and 7 are not adjacent to 14 in $\mathfrak{G}$, so it implies $1 \overset{H}{\sim} 11$.

III $1 \overset{H}{\sim} 7$: Assume otherwise and let $1 \overset{H}{\nsim} 7$, now since $1 \overset{\mathfrak{G}}{\sim} 7$ then $N_H(1) \subseteq (N_{\mathfrak{G}}(7) \cap N_{\mathfrak{G}}(1)) = \{5, 11, 12, 13\}$. Again according to part I and II,$1 \overset{H}{\sim} 11, 12$, therefore $N_H(1) = \{11, 12\}$. Here we have two possibilities, either $7 \overset{H}{\sim} 11$ or $7 \overset{H}{\sim} 12$. If $7 \overset{H}{\sim} 11$, since $14 \overset{\mathfrak{G}}{\nsim} 12$ then $14 \overset{H}{\sim} 11$ and this is a contradiction since $7 \overset{\mathfrak{G}}{\nsim} 14$. If $7 \overset{H}{\sim} 12$, since $2 \overset{\mathfrak{G}}{\nsim} 11$ then $2 \overset{H}{\sim} 12$ and this is a contradiction since $7 \overset{\mathfrak{G}}{\nsim} 2$. So it implies $1 \overset{H}{\sim} 7$.

IV $N_H(1) = \{7, 11, 12\}$: Since $\{7, 11, 12\}$ is a maximal clique in $\mathfrak{G}$, and $\{7, 11, 12\} \subseteq N_H(1)$ therefore $N_H(1) = \{7, 11, 12\}$.

□

Theorem 9. *Let $\mathfrak{G} = H^2$ for $g(H) = 5$, then H is either isomorphic to $\mathcal{G}_1$ or to $\mathcal{G}_2$.*

Proof. According to Lemma 6 we have:

- $N_H(12) = \{2, 3, 1\}$, since $L_1(2) = L_1(3) = 12$.

Also $14 \overset{H}{\sim} 11$ (because $L_1(14) = 11$), but $L_1(5) = L_1(13) = \{7, 11\}$, hence we have two possibilities:

I Case 1: $5 \overset{H}{\sim} 11$ and $13 \overset{H}{\sim} 7$:

- $N_H(7) = \{1, 11\}$: trivial.
- $N_H(11) = \{1, 5, 14\}$: trivial.
- $N_H(13) = N_G(7) - N_H(7) - \{11, 12\} = \{5, 6, 16\}$: since 13 and 1 are the only neighbours of 7.

  We now consider the set $N_{\mathfrak{G}}(13) - N_H(13) = \{1, 3, 8, 10, 11, 14, 15\}$, we have $L_{13}(3) = \{16\}$, $L_{13}(8) = \{6, 16\}$, $L_{13}(10) = \{16\}$, $L_{13}(14) = \{5, 6, 16\}$, $L_{13}(15) = \{5\}$, therefore:
- $N_H(5) = \{13, 15, 11\}$, since $14 \overset{\mathfrak{G}}{\sim} 11$ (otherwise we have a cycle of length four).
- $N_H(16) = \{13, 3, 8, 10\}$, since $14 \overset{\mathfrak{G}}{\nsim} 3$.
- $N_H(6) = \{13, 14\}$, trivial.

  We now consider the set $N_G(12) - N_H(12) = \{8, 9, 15, 16\}$, we have $L_{12}(8) = L_{12}(9) = L_{12}(15) = L_{12}(16) = \{2, 3\}$, however we know that $16 \overset{H}{\sim} 3, 8$:
- $N_H(2) = \{12, 15, 8\}$, since $16 \overset{H}{\sim} 8$ (otherwise we have a cycle of length four), and $9 \overset{H}{\nsim} 8$.

- $N_H(3) = \{12, 9, 16\}$, trivial.
- $N_H(15) = \{2, 4, 5, 9\}$, considering $N_{\mathfrak{G}}(5) - N_H(5)$ and refining the known neighbours.
- $N_H(14) = \{4, 6, 8, 11\}$, similar argument to vertex 15, considering the vertex 11.
- $N_H(4) = \{10, 14, 15\}$, similar argument to vertex 15, considering the vertex 15.
- $N_H(8) = \{2, 14, 16\}$, trivial.
- $N_H(9) = \{3, 15\}$, trivial.
- $N_H(10) = \{4, 16\}$, trivial.

  It can be seen that the above graph is $\mathcal{G}_1$.

II Case 2: $5 \overset{H}{\sim} 7$ and $13 \overset{H}{\sim} 11$:

- $N_H(7) = \{1, 5\}$: trivial.
- $N_H(11) = \{1, 13, 14\}$: trivial.
- $N_H(5) = N_G(7) - N_H(7) - \{1, 7\} = \{6, 7, 13, 16\}$: since 5 and 1 are the only neighbours of 7.

  We now consider the set $N_G(5) - N_H(5) = \{1, 2, 4, 9, 11, 14, 15\}$, we have $L_5(2) = \{16\}$, $L_5(4) = \{6, 16\}$, $L_5(9) = \{16\}$, $L_5(14) = \{6, 13, 16\}$, $L_5(15) = \{13\}$, therefore:
- $N_H(13) = \{5, 11, 15\}$, since $14 \overset{\mathfrak{G}}{\sim} 11$ (otherwise we have a cycle of length three).

- $N_H(16) = \{2, 4, 5, 9\}$, since $14 \overset{\mathfrak{G}}{\nsim} 9$, and also $4 \overset{\mathfrak{G}}{\nsim} 13$ but $4 \overset{\mathfrak{G}}{\sim} 11$, therefore $4 \overset{H}{\nsim} 6$.
- $N_H(6) = \{5, 14\}$, trivial. We now consider the set $N_G(12) - N_H(12) = \{8, 9, 15, 16\}$, we have $L_{12}(8) = L_{12}(9) = L_{12}(15) = L_{12}(16) = \{2, 3\}$, however we know that $16 \overset{H}{\sim} 3, 8$:
- $N_H(2) = \{12, 15, 8\}$, since $16 \overset{H}{\sim} 8$ (otherwise we have a cycle of length four), and $9 \overset{H}{\nsim} 8$.
- $N_H(3) = \{12, 9, 16\}$, trivial.
- $N_H(15) = \{3, 8, 10, 13\}$, considering $N_{\mathfrak{G}}(5) - N_H(5)$ and refining the known neighbours.
- $N_H(14) = \{4, 6, 8, 11\}$, similar argument to vertex 15, considering the vertex 11.
- $N_H(4) = \{10, 14, 16\}$, similar argument to vertex 15, considering the vertex 15.
- $N_H(8) = \{2, 14, 15\}$, trivial.
- $N_H(9) = \{3, 16\}$, trivial.
- $N_H(10) = \{4, 15\}$, trivial.

  It can be seen that the above graph is $\mathcal{G}_2$.

So H is either isomorphic to $\mathcal{G}_1$ or $\mathcal{G}_2$. □

4.3 Square of graphs with girth five

In this section we show that the following problem is NP-complete.

	Square of Graphs With Girth Five
Instance	A graph G.
Question:	Does there exists a graph H with girth at least 5 such that $G = H^2$?

It is an easy observation that Square of Graphs With Girth Five is in NP. We will reduce a variation of the "*positive 1-in-3 SAT*" problem (which is an NP-complete problem [32]) to Square of Graphs With Girth Five. Positive 1-in-3 SAT is a variant of the 3-satisfiability problem (3SAT). Like 3SAT, the input instance is a collection of clauses, where each clause is the disjunction of exactly three literals, and each literal is just a variable (there are no negations, which is why it is called positive). The positive 1-in-3 3SAT problem is to determine whether there exists a truth assignment to the variables so that each clause has exactly one true variable (and thus exactly two false variables). In this thesis we are interested in another variation of the positive 1-in-3 SAT, which we call Positive and Minimum Intersecting 1-in-3 SAT.

POSITIVE AND MINIMUM INTERSECTING 1-IN-3 SAT.

Instance: A collection of clauses, where each clause is the disjunction of exactly three variables and two different clauses are sharing at most one variable.

Question: Does there exists a truth assignment to the variables so that each clause has exactly one true variable?

Theorem 10. POSITIVE AND MINIMUM INTERSECTING 1-IN-3 SAT *is NP-complete.*

Proof. It is trivial that this problem is in NP. We reduce an instance of a *Positive 1-in-3 SAT* to a POSITIVE AND MINIMUM INTERSECTING 1-IN-3 SAT. Let ϕ be a given collection of clauses as an instance of the positive 1-in-3 SAT.

For each pair of clauses $c : (x \vee y \vee z)$ and $d : (x \vee y \vee u)$ in ϕ, that are sharing two variables x and y, we know u and v must have the same truth value. So we may identify the two variables and thus replace v with u and remove the clause d. We construct ϕ' from ϕ by removing one of the clauses in each pair of clauses that are sharing two variable. Therefore ϕ' is an instance of POSITIVE AND MINIMUM INTERSECTING 1-IN-3 SAT. This reduction shows that POSITIVE AND MINIMUM INTERSECTING 1-IN-3 SAT is NP-complete. □

In this section we reduce the POSITIVE AND MINIMUM INTERSECTING 1-IN-3 SAT to SQUARE OF GRAPHS WITH GIRTH FIVE.

4.3.1 The Reduction

Before introducing the reduction in full detail, we present three main ideas of the graph construction that we will explain below. For convenience, we represent $\forall a \in A : v \sim a$ by $v \sim A$, and also $\{x_a, y_a, z_a, \ldots\}$ by $\{x, y, z, \ldots\}_a$.

First is the idea of using graph $\mathfrak{G}$ to represent each copy of a variable. As we proved in Appendix A, a square root of $\mathfrak{G}$ is a graph which is isomorphic to either $\mathcal{G}_1$ or $\mathcal{G}_2$.

We set $\mathcal{G}_1$ to represent the FALSE value and $\mathcal{G}_2$ to represent the TRUE value. If the square root of the subgraph that is representing a copy of a variable x is isomorphic to $\mathcal{G}_1$ we conclude that x is $FALSE$. Otherwise, that is if it is isomorphic to $\mathcal{G}_2$, we conclude that x is $TRUE$.

The second idea is to represent a clause $c_i = x_i \vee y_i \vee z_i$ in such a way that exactly one of x_i, y_i and z_i is true (i.e., exactly one of the subgraphs that are representing the three variables is isomorphic to $\mathcal{G}_2$ and the other two are isomorphic to $\mathcal{G}_1$). For this, for each clause c_i we introduce four new vertices $y_1^i, \ldots, y_4^i$ to construct a Petersen graph in the square root (that is a K_{10} in the square graph) using vertices 5 and 13 in the three subgraphs representing the copies of variables in c_i. This construction is illustrated in Figure 4.9.

Lemma 7. *The square of the graph shown in Figure 4.9 has three different (up to labelling) square roots. The other two square roots can be obtained by switching $\mathcal{G}_1$s with $\mathcal{G}_2$s. However, it has a unique square root of girth 5 up to isomorphism.*

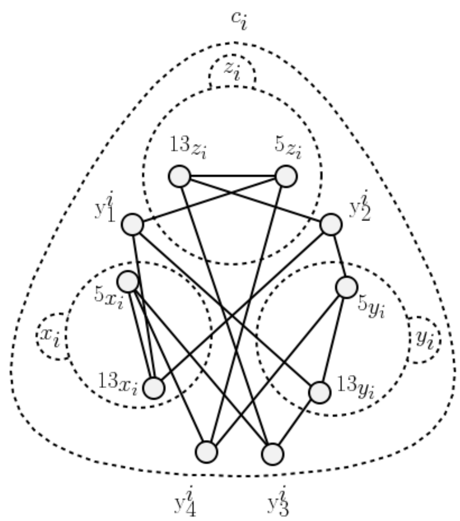

Figure 4.9: Structure of a subgraph of the square root which represents a clause.

Proof. Let $\mathfrak{C}$ be the square of the graph shown in Figure 4.9 on $X \cup Y \cup Z \cup I$ where $X = \{1, 2, \ldots, 16\}_{x_i}, Y = \{1, 2, \ldots, 16\}_{y_i}, Z = \{1, 2, \ldots, 16\}_{z_i}$ and $I = \{y_1^i, \ldots, y_4^i\}$. Also let D to be a square root of $\mathfrak{C}$. Graphs constructed by switching $\mathcal{G}_1$s with $\mathcal{G}_2$s. The isomorphism of these three graphs can be obtained by a permutation on I.

For example, assume that $D[X] \cong D[Y] \cong \mathcal{G}_1$ and $D[Z] \cong \mathcal{G}_2$. Then the graph obtained by the permutation $y_1^i \leftrightarrow y_3^i$ and $y_2^i \leftrightarrow y_4^i$ has the same square as the graph shown in Figure 4.9.

By Theorem 9, the square root of the subgraph induced by X, Y or Z is either $\mathcal{G}_1$ or $\mathcal{G}_2$. Now consider the neighbourhoods of vertices 5 and 13. We have $N_{\mathcal{G}_1}[5] = N_{\mathcal{G}_2}[13] = \{5, 13, 11, 15\}$ and $N_{\mathcal{G}_1}[13] = N_{\mathcal{G}_2}[5] = \{5, 13, 6, 7, 16\}$. It can be seen that if none or more than one of the square roots of the subgraph induced by X, Y or Z is isomorphic to $\mathcal{G}_2$, then there would be no permutation on I, that form the same square as the graph shown in Figure 4.9. $\square$

The third idea is to make sure that different copies of the same variable have the same truth value. Again we use the fact that $N_{\mathcal{G}_1}[5] = N_{\mathcal{G}_2}[13] = \{5, 13, 11, 15\}$ and $N_{\mathcal{G}_1}[13] = N_{\mathcal{G}_2}[5] = \{5, 13, 6, 7, 16\}$. Let x_i and x_j be two copies of the same variable in two different clauses c_i and c_j. We introduce two new vertices called v_{x_i,x_j} and w_{x_i,x_j} which form a C_6 in the square root together with the vertices 5 and 13 in the subgraphs corresponding to x_i and x_j. If both x_i and x_j are TRUE then $v_{x_i,x_j} \sim \{13_{x_i}, 13_{x_j}\}$ and $w_{x_i,x_j} \sim \{5_{x_i}, 5_{x_j}\}$, otherwise $w_{x_i,x_j} \sim \{13_{x_i}, 13_{x_j}\}$ and $v_{x_i,x_j} \sim \{5_{x_i}, 5_{x_j}\}$. This construction is shown in Figure 4.10. Moreover we have the following Lemma.

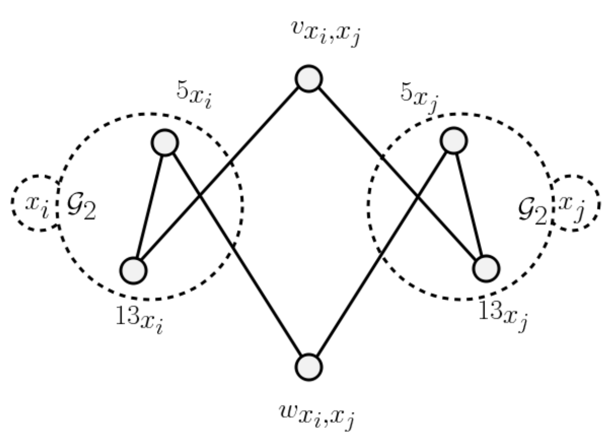

Figure 4.10: Different copies of a variable have the same truth value.

Lemma 8. *Let $\mathfrak{X}$ be the square of the graph shown in Figure 4.10 on the vertex set of $X^i \cup X^j \cup \{v_{x_i,x_j}, w_{x_i,x_j}\}$ where $X^i = \{1, 2, \ldots, 16\}_{x_i}$ and $X^j = \{1, 2, \ldots, 16\}_{x_j}$. Let $\mathsf{X}^2 = \mathfrak{X}$. If $\mathsf{X}[X^i] \cong \mathcal{G}_1$ (or $\mathcal{G}_2$) then $\mathsf{X}[X^j] \cong \mathcal{G}_1$ (or $\mathcal{G}_2$).*

Proof. Assume otherwise and let (without loss of generality) $\mathsf{X}[X^i] \cong \mathcal{G}_1$ while $\mathsf{X}[X^j] \cong \mathcal{G}_2$, hence v_{x_i,x_j} must be adjacent to 5_{x_i} and 13_{x_j} which means $5_{x_i} \overset{\mathfrak{X}}{\sim} 13_{x_j}$, and this is a contradiction as $5_{x_i} \overset{\mathfrak{X}}{\nsim} 13_{x_j}$. $\square$

Reduction Graph: Let $\phi : (c_1 \wedge c_2 \wedge \cdots \wedge c_n)$ be an instance of Positive and Minimum Intersecting 1-in-3 SAT such that $c_i = x_i \vee y_i \vee z_i$. As a convention we use x_i and x_j to represent two copies of variable x in distinct clauses c_i and c_j.

We construct an instance $G = G(\phi)$ and we show that there exists a square root H of girth 5 of graph G that corresponds to a satisfying assignment of ϕ.

The vertex set of graph $G(\phi)$ consists of:

- For every copy x_i of variable x, $V_{x_i} = V_{\mathcal{G}_1}(= V_{\mathcal{G}_2}) = \{1, 2, \ldots, 16\}_{x_i}$, representing 16 vertices of a graph $\mathfrak{G}$.
- For each clause c_i, $V_i = \{y^i_1, y^i_2, y^i_3, y^i_4\}$.
- $W_{x_i,x_j} = \{v_{x_i,x_j}, w_{x_i,x_j}\}$, corresponding to two copies x_i and x_j of the same variable x, in two distinct clauses c_i and c_j.

The edge set of $G(\phi)$ consists of:

- Variable edges: for each x_i, $G[V_{x_i}] = \mathfrak{G}$.

- Clause edges: For each clause $c_i = x_i \vee y_i \vee z_i$

  $G[\{5_{x_i}, 13_{x_i}, 5_{y_i}, 13_{y_i}, 5_{z_i}, 13_{z_i}, y_1^i, y_2^i, y_3^i, y_4^i\}] \cong K_{10}$, i.e., they are all adjacent to each other. Also by recalling that $N_{\mathcal{G}_1}[5] = N_{\mathcal{G}_2}[13] = \{5, 13, 11, 15\}$ and $N_{\mathcal{G}_1}[13] = N_{\mathcal{G}_2}[5] = \{5, 13, 6, 7, 16\}$, we have:

  $y_1^i \sim \{11_{x_i}, 15_{x_i}, 6_{y_i}, 7_{y_i}, 16_{y_i}, 11_{z_i}, 15_{z_i}\}$,

  $y_2^i \sim \{6_{x_i}, 7_{x_i}, 16_{x_i}, 11_{y_i}, 15_{y_i}, 11_{z_i}, 15_{z_i}\}$,

  $y_3^i \sim \{6_{x_i}, 7_{x_i}, 16_{x_i}, 6_{y_i}, 7_{y_i}, 16_{y_i}, 6_{z_i}, 7_{z_i}, 16_{z_i}\}$,

  $y_4^i \sim \{11_{x_i}, 15_{x_i}, 6_{y_i}, 7_{y_i}, 16_{y_i}, 6_{z_i}, 7_{z_i}, 16_{z_i}\}$, see Figure 4.11.

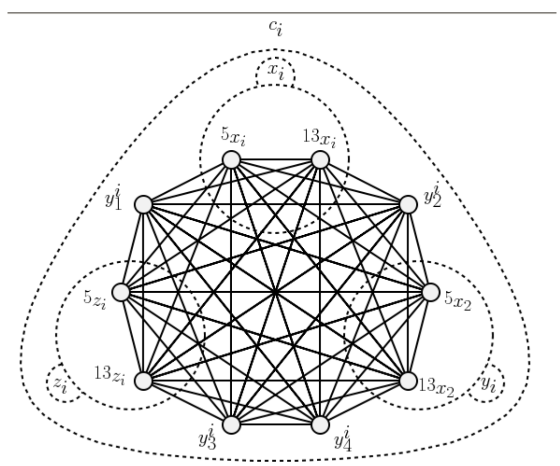

Figure 4.11: A subgraph of $G(\phi)$ corresponding to a clause.

- Intra clause edges: for each clause $c_i = x_i \vee y_i \vee z_i$ where $i \notin \{j, k, m\}$:

  $y_1^i \sim \{v_{x_i,x_j}, w_{y_i,y_k}, v_{z_i,z_m}\}$,

  $y_2^i \sim \{w_{x_i,x_j}, v_{y_i,y_k}, v_{z_i,z_m}\}$,

  $y_3^i \sim \{w_{x_i,x_j}, w_{y_i,y_k}, w_{z_i,z_m}\}$,

$y_4^i \sim \{v_{x_i,x_j}, v_{y_i,y_k}, w_{z_i,z_m}\}$.

Notice that we may have only a subset of these edges depending on the existence of x_j (the copy of variable x in c_j), y_k (the copy of variable y in c_k) and z_m (the copy of variable z in c_m).

- Edges for different copies of a variable: for each arbitrary pair x_i and x_j which are different copies of the same variable,

  $v_{x_i,x_j} \sim \{13_{x_i}, 13_{x_j}, 5_{x_i}, 5_{x_j}\}$,

  $v_{x_i,x_j} \sim \{11_{x_i}, 15_{x_i}, 11_{x_j}, 15_{x_j}\}$,

  $w_{x_i,x_j} \sim \{13_{x_i}, 13_{x_j}, 5_{x_i}, 5_{x_j}\}$,

  $w_{x_i,x_j} \sim \{6_{x_i}, 7_{x_i}, 16_{x_i}, 6_{x_j}, 7_{x_j}, 16_{x_j}\}$, see Figure 4.12.

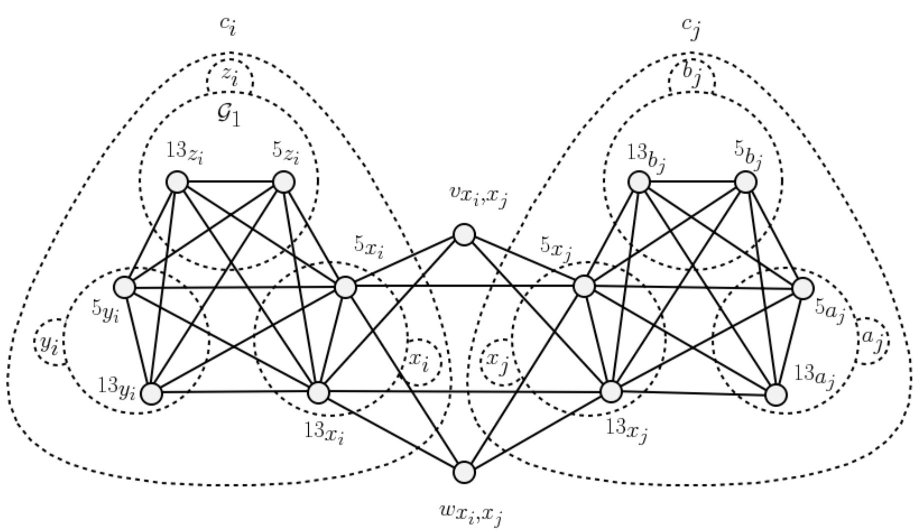

Figure 4.12: A subgraph of $G(\phi)$ corresponding to the clause $c_i = x_i \vee y_i \vee z_i$.

- Edges of variable copies:

  for an arbitrary variable x and all $i \neq j$ and $k \neq l$, we have $w_{x_i,x_j} \sim w_{x_k,x_l}$ and $v_{x_i,x_j} \sim v_{x_k,x_l}$.

It is an easy observation to see that $G(\phi)$ can be constructed from ϕ in polynomial time.

Lemma 9. *There exists a truth assignment to variables in instance ϕ of POSITIVE AND MINIMUM INTERSECTING 1-in-3 SAT that satisfies the formula if and only if there exists a graph H of girth five such that $G(\phi) = H^2$.*

Proof.
- *Satisfiability to squareness:*
  - H construction: we construct the graph H by using a satisfying assignment of ϕ as follows:
    - For all i such that there exists a clause c_i where $x_i \in c_i$, $H[\{1, 2, \ldots, 16\}_{x_i}] = \mathcal{G}_2$ if x is true and $H[\{1, 2, \ldots, 16\}_{x_i}] = \mathcal{G}_1$ if x is false.
    - For each pair of x_i and x_j where $i \neq j$ if x is true then $v_{x_i,x_j} \overset{H}{\sim} \{13_{x_i}, 13_{x_j}\}$ and $w_{x_i,x_j} \overset{H}{\sim} \{5_{x_i}, 5_{x_j}\}$. Otherwise, that is if x is false, $w_{x_i,x_j} \overset{H}{\sim} \{13_{x_i}, 13_{x_j}\}$ and $v_{x_i,x_j} \overset{H}{\sim} \{5_{x_i}, 5_{x_j}\}$.
    - For each clause $c_i = x_i \vee y_i \vee z_i$:

      if x_i is true then

      $y^i_1 \overset{H}{\sim} \{13_{x_i}, 13_{y_i}, 5_{z_i}\}$, $y^i_2 \overset{H}{\sim} \{5_{x_i}, 5_{y_i}, 5_{z_i}\}$, $y^i_3 \overset{H}{\sim} \{5_{x_i}, 13_{y_i}, 13_{z_i}\}$,

$y_4^i \overset{H}{\sim} \{13_{x_i}, 5_{y_i}, 13_{z_i}\}$.

if y_i is true then
$y_1^i \overset{H}{\sim} \{5_{x_i}, 5_{y_i}, 5_{z_i}\}$, $y_2^i \overset{H}{\sim} \{13_{x_i}, 13_{y_i}, 5_{z_i}\}$, $y_3^i \overset{H}{\sim} \{13_{x_i}, 5_{y_i}, 13_{z_i}\}$,
$y_4^i \overset{H}{\sim} \{5_{x_i}, 13_{y_i}, 13_{z_i}\}$.

if z_i is true then
$y_1^i \overset{H}{\sim} \{5_{x_i}, 13_{y_i}, 13_{z_i}\}$, $y_2^i \overset{H}{\sim} \{13_{x_i}, 5_{y_i}, 13_{z_i}\}$, $y_3^i \overset{H}{\sim} \{13_{x_i}, 13_{y_i}, 5_{z_i}\}$,
$y_4^i \overset{H}{\sim} \{5_{x_i}, 5_{y_i}, 5_{z_i}\}$.

Recall that in all cases 10 vertices y_i^1, y_i^2, y_i^3, y_i^4, 5_{x_i}, 5_{y_i}, 5_{z_i}, 13_{x_i}, 13_{y_i} and 13_{z_i} form a Petersen graph in H.

- $H^2 = G(\phi)$: trivial.

• *Squareness to satisfiability:*
Let H be a square root of $G(\phi)$. By Theorem 9, graph $H[V_{x_i}]$ (for each copy of an arbitrary x) is isomorphic either to $\mathcal{G}_1$ or $\mathcal{G}_2$. We set x to be true when $H[V_{x_i}] \cong \mathcal{G}_2$ and false otherwise. By Lemma 8 all other copies of x would also have the same truth value. By Lemma 7 this assignment is a truth assignment to ϕ since exactly one variable in each clause is evaluated as true.

□

As an example let $\phi : c_1 \wedge c_2 \wedge c_3$ and $c_1 = x_1 \vee y_1 \vee z_1$, $c_2 = x_2 \vee u_2 \vee v_2$ and

$c_3 = y_3 \vee a_3 \vee b_3$, where $x = b = TRUE$ and $y = z = u = a = v = FALSE$.
The graph shown in Figure 4.13 is the square root of $G(\phi)$.

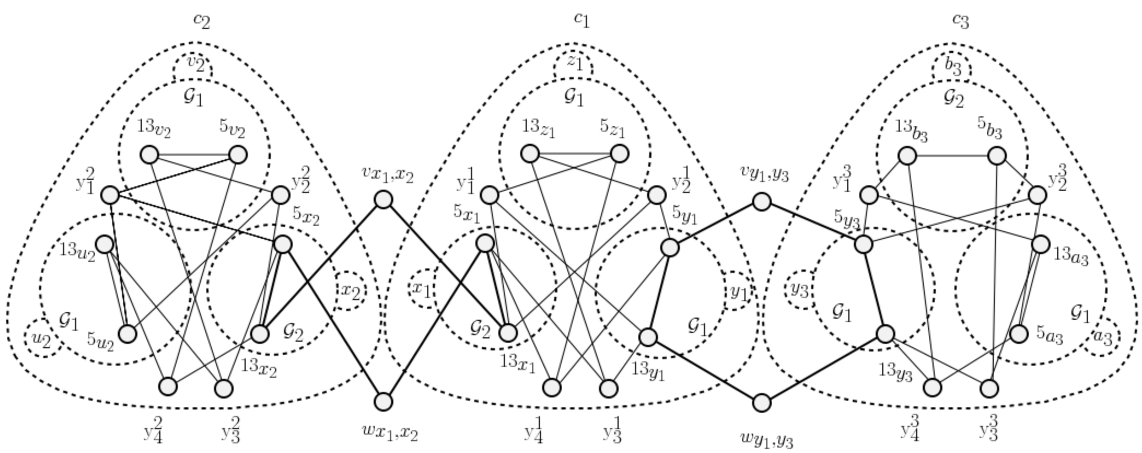

Figure 4.13: An example of H.

Theorem 11. Square of Graphs With Girth Five *is NP-complete.*

Theorem 12 (The Complete Dichotomy Theorem)**.** Square of Graphs With Girth g *is NP-complete if and only if* $g \leq 5$.

Chapter 5

Conclusion

We proved that Square of Graphs With Girth Five is NP-complete. Together with results provided by Motwani and Sudan [23] and Farzad et al. [11], we presented Theorem 12 as a complete dichotomy theorem for the square root finding problem. We also found an efficient algorithm when the square root has girth five but excludes specific dense subgraphs. With respect to uniqueness of the square root, we present graphs that have exponentially many square roots of girth five.

To continue this line of research one may follow some natural extensions of the problem. Bound the number of non-isomorphic square roots with girth 5 for a given graph based on n (size of the graph). We showed that there are graphs with many non-isomorphic square roots with girth five. A bound on the number of non-isomorphic square roots can be useful to determine if there exists an efficient algorithm for root finding in specific graphs. Such bound can be beneficial to answer the problem of finding a square root for K_{57^2+1}.

However the problem of square root finding for graphs can be defined for higher roots.

	k^{th} Power of a Graph With Girth r
Instance:	A graph G.
Question:	Does there exists a graph H with girth at least r such that $G = H^k$.

The problem of root finding for higher root is an open problem in terms of the r^{th}-root of the power graph. Results provided by Adamaszek and Adamaszek [2] is the closest result to a complete girth-parametrized complexity dichotomy. They proved that the recognition problem of k^{th}Power of a Graph With Girth r is NP-complete when $r = k$ while there is a polynomial time algorithm to find all k^{th}-roots of girth $2k + 3$ for a given graph.

The problem of finding a complete girth-parametrized complexity dichotomy for k^{th}Power of a Graph With Girth r is open and we conjectured the following:

Conjecture 1. k^{th}Power of a Graph With Girth r *for* $r = 2k + 1$ *is NP-complete.*

Another area of interest in this research is the cross field work of root finding and the study of *testing graph algorithms.* As we advance further in the the study of efficient computing, we would face different variations of "efficiency" for algorithms. For example see [25] for a different definition of an algorithm. Usually, efficiency is represented with a polynomial time complexity (with unrestricted access to the input). Recent definitions of efficiency are stricter in time as well as the access to the input. In new frameworks we have to answer a decision problem by using a constant number of calls to little pieces (ideally random) of our input. Such probabilistic algorithms are called *testing algorithms*. Testing algorithms in Graph Theory is called *Graph Testing*, and is one of the main source of interests in this area.

Locality in Graph Algorithms is a new approach in graph testing. Roughly speaking, the problem of locality is as follows. In a distributed network (without shared memory), processors are exchanging their messages. Since sending messages to far away nodes is expensive, computation should be based on local information as much as possible, i.e. every vertex is only sending messages to its neighbours. There are many amazing properties of a graph that an algorithm can decide by using only local informations, but also there are some challenging questions. The question is for what properties of the graph can locality be achieved?

Results in root finding can be extended to the study of locality in distributed graphs, as these two concepts are fundamentally connected. In particular root finding can be a useful test to decide the properties that can or can not be solved by using local information of graphs. Our approach is intuitively applicable in basic results of locality as we restate some results in locality. A basic interconnection is the NP-completeness of deciding (with local information) whether a graph contains a square or not, which can be restated using the NP-completeness of "recognition of square of graphs with girth four".

Bibliography

[1] A. Adamaszek, M. Adamaszek Uniqueness of Graph Square Roots of Girth Six, *Electr. J. Comb.* 18(1): (2011).

[2] A. Adamaszek, M. Adamaszek, Large-girth roots of graphs, *SIAM J. Discrete Math.* 24 (4) (2010) 1501–1514.

[3] N. Linial, Locality in distributed graph algorithms, *SIAM J. on Computing* 21 (1992) 193–201.

[4] G. Agnarsson, R. Greenlaw, M. M. Halldórsson, On powers of chordal graphs and their colorings, *Congressus Numer.* 144 (2000) 41–65.

[5] N. Alon, B. Mohar, The chromatic number of graph powers, *Combinatorics, Probability and Computing* 11 (2002) 1–10.

[6] A. Brandstädt, V. B. Le, and R. Sritharan, Structure and linear time recognition of 4-leaf powers, *ACM Transactions on Algorithms*, to appear.

[7] M. Chang, M. Ko, and Hsueh-I Lu, Linear time algorithms for tree root problems, *Lecture Notes in Computer Science*, 4059 (2006) 411–422.

[8] D. W. Cranston, S. J. Kim, List-coloring the square of a subcubic graph, *J. Graph Theory* 57 (2007) 65–87.

[9] E. Dahlhaus, P. Duchet, On strongly chordal graphs, *Ars Combin.* 24 B (1987) 23–30.

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

[11] B. Farzad, L. C. Lau, V. B. Le, N. N. Tuy, Computing Graph Roots Without Short Cycles *Proc. 26th STACS* (2009) 397-408.

[12] M. R. Garey, D. S. Johnson, *Computers and Intractability–A Guide to the Theory of NP-Completeness*, Freeman, New York (1979), twenty-third printing 2002.

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

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

[15] F. Havet, Choosability of the square of planar subcubic graphs with large girth, *Discrete Math.*, to appear.

[16] J.D. Horton, K. Kilakos, Minimum edge dominating sets, *SIAM J. Discrete Math.* 6 (1993) 375–387.

[17] A. Itai, M. Rodeh, Finding a minimum circuit in a graph, *SIAM J. Computing* 7 (1978) 413–423.

[18] P. E. Kearney, D. G. Corneil, Tree powers, *J. Algorithms* 29 (1998) 111–131.

[19] L. C. Lau, Bipartite roots of graphs, *ACM Transactions on Algorithms* 2 (2006) 178–208.

[20] L. C. Lau, D. G. Corneil, Recognizing powers of proper interval, split and chordal graphs, *SIAM J. Discrete Math.* 18 (2004) 83–102.

[21] Y. Lin, S. S. Skiena, Algorithms for square roots of graphs, *SIAM J. Discrete Math.* 8 (1995) 99–118.

[22] A. Lubiw, Γ-free matrices, *Master Thesis*, Dept. of Combinatorics and Optimization, University of Waterloo, Canada, 1982.

[23] R. Motwani, M. Sudan, Computing roots of graphs is hard, *Discrete Appl. Math.* 54 (1994) 81-88.

[24] A. Mukhopadhyay, The square root of a graph, *J. Combin. Theory* 2 (1967) 290-295.

[25] C. H. Papadimitriou, Computational Complexity, *Addison Wesley* (1994).

[26] A. Raychaudhuri, On powers of strongly chordal and circular arc graphs, *Ars Combin.* 34 (1992) 147–160.

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

[28] R. R. Singleton, There is no irregular Moore graph, *American Mathematical Monthly* 75(1) (1968) 42–43.

[29] S. Tsukiyama, M. Ide, H. Ariyoshi, and I. Shirakawa, A new algorithm for generating all the maximal independent sets, *SIAM J. Computing* 6 (1977) 505–517.

[30] D. B. West, Introduction to Graph Theory - Second edition, *Prentice Hall* (2001).

[31] R. Karp, Reducibility Among Combinatorial Problems, R. E. Miller and J. W. Thatcher (editors). Complexity of Computer Computations. *New York: Plenum.* (1972) 85–103.

[32] T. J. Schaefer, The complexity of satisfiability problems, *In Proceedings of the* 10^{th} *Annual ACM Symposium on Theory of Computing (New York), ACM, New York* (1978) 126–226.

Appendices

Appendix A

Input: A graph G, a vertex $v \in V_G$ and a subset $U \subseteq N_G(v)$.

Output: A graph H of girth more than six with $H^2 = G$ and $N_H(v) = U$ or else 'NO' if such a square root H of G does not exist.

Add all edges vu, $u \in U$, to E_H
$Q \leftarrow \emptyset$
    for all $u \in U$ **do**
        enqueue u onto Q
        parent$(u) \leftarrow v$
    while Q is not empty **do**
        $u \leftarrow$ dequeue(Q)
        set $W := N_G(u) \cap \big(N_G(\text{parent}(u)) \setminus N_H(\text{parent}(u))\big)$
        for each $w \in W$ **do**
            add uw to E_H
        if parent$(w) = \emptyset$ **then:**
            parent$(w) \leftarrow u$
            enqueue(Q, w)
     else
          return(NO)

Appendix B

Input: A graph G, a vertex v and a neighbourhood $U = N_G(v)$

Output: $H = \sqrt{G}$ such that $H \in \mathcal{H}_1$ with $N_H(v) = U$ or else NO if such square root does not exists.

Add all edges vu, $u \in U$, to E_H:
    for all $u \in U$ **do**
        enqueue u onto Q
        parent(u)$\leftarrow v$
    while Q is not empty **do**
        $u \leftarrow Q$.dequeue()
        set v :=parent(u)
    for all $x \in (N_G(v) - N_H(v)) \cap N_G(u)$ **do**
        enqueue x onto S
        while S is not empty **do**
        $x \leftarrow S$.dequeue()
            if parent(x)$\neq \emptyset$ **then:**
                set $L_v(x) = N_G(x) \cap N_H(v)$
                if $|L_v(x)| = 1$ **then:**
                    set u := parent(x)
                    $E_H := E_H \cup \{ux\}$

else then:

set $y \in (N_G(u) - N_G(u') \cap (N_G(v) - N_H(v)))$

if $yx \in E_G$ **then:**

set $u :=$ parent(x)

$E_H := E_H \cup \{ux\}$

else then:

set $u' =$ parent(x)

$E_H := E_H \cup \{u'x\}$

enqueue(Q, x)

if $G = H^2$ **then**

return(H)

else

return(NO)

Appendix C

Input: A graph G, a vertex v and a neighbourhood $U = N_G(v)$

Output: $H = \sqrt{G}$ such that $H \in \mathcal{H}_1$ with $N_H(v) = U$ or else NO if such square root does not exists.

Add all edges vu, $u \in U$, to E_H:
    for all $u \in U$ **do**
        enqueue u onto Q
        parent(u)$\leftarrow v$
    while Q is not empty **do**
        $u \leftarrow Q$.dequeue()
        set v :=parent(u)
    for all $x \in (N_G(v) - N_H(v)) \cap N_G(u)$ **do**
        enqueue x onto S
        while S is not empty **do**
        $x \leftarrow S$.dequeue()
            if parent(x)$\neq \emptyset$ **then:**
                set $L_v(x) = N_G(x) \cap N_H(v)$
                if $|L_v(x)| = 1$ **then:**
                    set u := parent(x)
                    $E_H := E_H \cup \{ux\}$

else, if $|L_v(x)| = 3$ **then:**

set $W := \{y \in N_G(v) - N_H(v) | L_v(x) \cap L_v(y)| \geq 2\}$

set $w := \bigcap_{y \in W} L_v(y)$

set $w :=$ parent(x)

$E_H := E_H \cup \{wx\}$

for all $y \in W$ **do**

set $h := L_v(y) - \{w\}$

set $h :=$ parent(y)

$E_H := E_H \cup \{hy\}$

else, if $|L_v(x)| = 2$ **then:**

set $y \in (N_G(u) - N_G(u') \cap (N_G(v) - N_H(v)))$

if $yx \in E_G$ **then:**

set $u :=$ parent(x)

$E_H := E_H \cup \{ux\}$

else then:

set $u' =$ parent(x)

$E_H := E_H \cup \{u'x\}$

enqueue(Q, x) .

if $G = H^2$ **then**

return(H)

else

return(NO)

List of Figures