University of Alberta

STRICTLY CHORDAL GRAPHS AND PHYLOGENETIC ROOTS

by

William Kennedy

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Computing Science

Edmonton, Alberta
Fall 2005

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Strictly Chordal Graphs and Phylogenetic Roots** submitted by William Kennedy in partial fulfillment of the requirements for the degree of **Master of Science**.

____

Ryan B. Hayward
Supervisor

____

Guohui Lin
Co-Supervisor

____

Piotr Rudnicki

____

René Poliquin
External Examiner

Date: ____________________

Abstract

A phylogeny is the evolutionary history for a set of evolutionarily related species. The development of hereditary trees, or phylogenetic trees, is an important research subject in computational biology. One development approach, motivated by graph theory, constructs a relationship graph based on evolutionary proximity of pairs of species. Associated with this approach is the kth phylogenetic root construction problem: given a relationship graph, construct a phylogenetic tree such that the leaves of the tree correspond to the species and are within distance k if adjacent in the relationship graph. In this thesis, we give a polynomial time algorithm to solve this problem for strictly chordal graphs, a particular subclass of chordal graphs. During the construction of a solution, we examine the problem for tree chordal graphs, and establish new properties for strictly chordal graphs.

Acknowledgements

First, I would like to thank Ryan Hayward for approaching me after his algorithms lecture to offer me a job as his research assistant; this event has had a great impact on the direction of my studies. His continuing support and guidance as my supervisor has been greatly appreciated. I would also like to thank Guohui Lin for introducing me to Computation Biology and for his guidance in developing the ideas of this thesis. I am grateful to Bill Rosgen and Jim Nastos for being sounding boards for my many ideas and for sharing their own algorithmic problems with me. I would like to thank my family for their continuing support in all that I pursue, academic and otherwise. Finally, I would like to thank Amy for her love and support.

Table of Contents

List of Figures

List of Symbols

Let A and B be sets. Let G, G_1 and G_2 be graphs. Let T be a tree. Let $\mathcal{H}$ be a hypergraph. Let m be a maximal clique and c_1 and c_2 be two critical cliques. Let i and k be integers.

$\|A\|$	The number of elements in A.
$A \subseteq B$	A is a *subset* of B.
$A \subset B$	A is a *proper subset* of B.
$A \cap B$	The *intersection* of A and B.
$A \cup B$	The *union* of A and B.
$A \setminus B$	The *set difference* of B from A.
$\emptyset$	The *empty set*.
$d_T(a, b)$	The *distance* in T from a to b.
$A \times B$	The *Cartesian product* of A and B.
$V(G)$	The *vertex set* of G.
$E(G)$	The *edge set* of G.
$N_G(v)$	The *neighbourhood* of vertex v in G.
$deg_G(V)$	The *degree* of vertex v in G.
P_n	A *chordless path* on n vertices.
C_n	A *chordless cycle* on n vertices.
K_n	A *clique* on n vertices.
cccard(m)	The *critical clique cardinality* of m.
$G[W]$	The *subgraph of* G *induced* by $W \subseteq V(G)$.
$G_1 \cong G_2$	G_1 is isomorphic to G_2.
G^k	The *k-th power* of G.
$V(\mathcal{H})$	The *vertex set* of $\mathcal{H}$.
$\mathcal{E}(\mathcal{H})$	The *hyperedge set* of $\mathcal{H}$.
$CC(G)$	The *critical clique graph* of G.
$i(mod\ j)$	i modulo j.
$(G_1, c_1) \star (G_2, c_2)$	The *clique join* of pairs (G_1, c_1) and (G_2, c_2).

Chapter 1

Introduction

1.1 Historical Background

1.1.1 Phylogenetics

The evolutionary history of a set of evolutionary units (for example, organisms) is its *phylogeny*; a hereditary tree which represents this history is its *phylogenetic tree*. Such a tree can be used to describe the pattern and timing of branching events in the evolution of a group of units. Some important applications of phylogenetic trees are determining the closest evolutionary relatives of an organism and determining the function and origin of genes.

The study of evolutionary relationships belongs to the biological subject of *phylogenetics*. Phylogenetic analysis, a method of deducing these evolutionary relationships, consists of four major steps [1]: alignment, determining the substitution model, tree building, and tree evaluation. Starting with appropriate character sequences corresponding to each evolutionary unit, alignment is a means for describing how related these sequence are and produces the data set used in the model of evolution. The substitution model of evolution describes the probability of a difference between characters, found in an alignment, occurring. Construction of phylogenetic trees use the substitution model of evolution and an alignment to deduce a hereditary tree. Finally, tree evaluation calculates the probability of the tree being representative of the data. The books [33, 1, 16] discuss each of these four steps in detail. The problem considered in this thesis is inspired from the third of these four steps — phylogenetic tree construction.

By the Darwinian school of thought, a central idea to the development of a phylogenetic tree is that all life forms have descended from a common ancestor [26]. By this assumption it follows that there exists a historically accurate tree, or *true tree*, for the evolution of a set of evolutionary units. An *inferred tree* is a tree constructed based on the particular phylogenetic analysis method used. The inferred tree is an estimate of the genetic connection between evolutionary units and the chronological spacing of branching events in their evolution.

A *tree* is an acyclic connected graph. Roughly speaking, a *phylogenetic tree* is a tree whose nodes are either *external leaf* nodes representing the evolutionary units or *internal* nodes representing ancestors. A tree is *bifurcating* if all internal nodes have degree three.

A phylogenetic tree can be rooted or unrooted. A rooted tree represents the evolution of a group of evolutionary units where the root represents the common ancestor. An unrooted tree shows only evolutionary proximity of units. If an unrooted tree is considered as a rooted tree whose root is unknown then a root can be found using the outgroup method [30], though this rooted tree is not always accurate (with respect to the true tree). Both types of phylogenetic trees are useful in understanding a phylogeny [1].

With respect to a unit set $\Omega_n = \{\omega_1, \omega_2, ..., \omega_n\}$ and an alignment, a *dissimilarity matrix* $D = d_{ij}$ for Ω_n is a matrix whose entries d_{ij} give the alignment score between units ω_i and ω_j, where the closeness of units is indicated by the smallness of the score. With respect to a tree T with leaves labelled by the set Ω_n, $\pi(T)$ is the path length matrix where entries $\pi(T)_{ij}$ give the distance in T between ω_i and ω_j. In the construction of an evolutionary tree T based on the dissimilarity matrix D, the goal is to find such a tree T so that $\pi(T)$ fits well with D. One measure of fitness is the *goodness-of-fit* [12] between D and $\pi(T)$ as

$$F_\alpha(D, \pi(T)) = \sum_{1 \le i < j \le n} |d_{ij} - \pi(T)_{ij}|^\alpha.$$

An *additive tree* [26] is a tree with leaves labelled by the set Ω_n and with positive real edge weights such that the sum of edge weights between leaves labelled by ω_i and ω_j is d_{ij}.

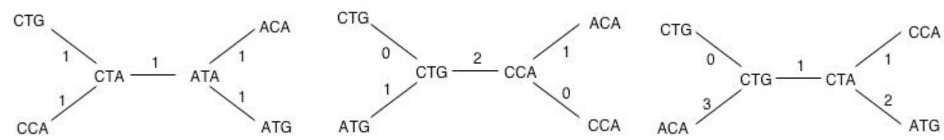
FITTING ADDITIVE TREES [12]
Instance: Set $\Omega_n, n \ge 3$; matrix D; positive integer k.
Query: Does an additive tree T exist such that $F_\alpha(D, \pi(T)) \le k$?

Day [12] showed that FITTING ADDITIVE TREES is NP-complete for $\alpha \in \{1, 2\}$. This is not surprising, as the number of additive trees grows exponentially in the number of units in Ω_n. For example, the number of unrooted bifurcating trees with n elements is $(2n-5)!/[(n-3)!2^{n-3}]$ [14].

1.1.2 Construction Techniques of Phylogenetic Trees

Two methods of phylogenetic root construction commonly used in practice are either to limit the search using heuristics to find a probable tree, or to find the optimal tree by searching all possibilities, which is often computationally expensive. The goal of such construction is to take the information derived for a set of evolutionary units through alignment and the substitution model and produce an inferred tree that is closest to the true tree. Phylogenetic tree construction algorithms belong to two major categories: character-based methods and distance-based methods. Character-based methods search for the optimal tree based on character pattern differences between the species, whereas distance-based methods use calculated evolutionary distance to construct a tree.

The *maximum parsimony* [26] technique is a character-based method that finds the tree that minimizes the total evolutionary change that has occurred between elements of the group. As the word parsimony suggests, the basis of this method is the assumption that a simpler explanation is more desirable than a complicated one. As such algorithms must search through the exponentially many unrooted trees, this method becomes computationally infeasible for a large number of evolutionary units. See Figure 1.1 for an example.



(a) Plausible tree with 5 character replacements.

(b) Plausible tree with 5 character replacements.

(c) Plausible tree with 7 character replacements.

Figure 1.1: The three possible unrooted trees using the maximum parsimony technique given four character sequences: {CTG, ATG, ACA, CCA}. Either tree (a) or (b) will be chosen as they both have the minimum number of nucleotide replacements.

The *maximum likelihood* [26] method is a character-based method that finds the tree with the highest probability of occurrence from the given data. Based on a model of assumptions for the probability of an event occurring, such as genetic mutation, this method searches for the tree with the highest likelihood of existence. Again, the main disadvantage of this method is that current methods search the exponentially many possible phylogenetic trees, making this technique computationally expensive and therefore infeasible for sets with a large number of evolutionary units.

The following two distance-based methods attempt to construct the tree by considering 'neighbours' in an evolutionary tree, where units of a subtree are considered neighbours of units of another subtree if the roots of the two subtrees are siblings in the evolutionary tree. See Figure 1.2.

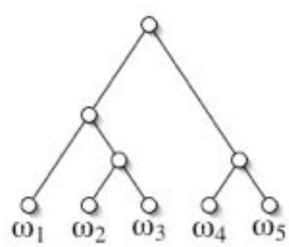


Figure 1.2: Neighbours in an evolutionary tree. ω_1 is an evolutionary neighbour of $\{\omega_2, \omega_3\}$. ω_2 is an evolutionary neighbour of ω_3. ω_4 is an evolutionary neighbour of ω_5. $\{\omega_1, \omega_2\ \omega_3\}$ is an evolutionary neighbour of $\{\omega_4, \omega_5\}$.

The *neighbour-relation* [36] method uses the insight called the *four-point condition* [26], which states that for an additive tree T where a and b are neighbours and c and d are neighbours, then

$$d_T(a, b) + d_T(c, d) \leq d_T(a, c) + d_T(b, d).$$

The algorithm searches all sets of four elements and chooses the pair of neighbours that most frequently satisfy this condition with the other elements in the tree. This pair of elements becomes neighbours in the tree and is subsequently viewed as a single element. The algorithm continues until all units have been processed.

The *neighbour-joining* [35] method is a greedy algorithm that iteratively minimizes total branch length. Starting with all elements adjacent to a central node, this method looks for the closest neighbours and creates a new branch with its distance equal to the mean of its two elements.

The focus of this thesis is an algorithm that is a graph theoretic variation of the distance method – the *kth phylogenetic root construction*. Given a dissimilarity matrix for a set of evolutionary units Ω_n, our algorithm creates an input graph G by making the units adjacent if their dissimilarity distance is under a given threshold. It then creates a tree T where the vertices of G correspond to the leaves of T where two vertices of G are adjacent if within a given path-length distance in T. Formally, given a dissimilarity matrix D and a threshold t, construct a new matrix D' such that:

$$D'_{ij} = \begin{cases} 1 & if\ D_{ij} \leq t \\ 0 & \text{otherwise.} \end{cases}$$

Using D' as the input graph, our algorithm constructs a corresponding tree where leaves are labelled by the units, and, for a fixed positive integer k, units ω_i and ω_j are within distance k if and only if $D'_{ij} = 1$. Before we discuss our algorithm further, we first introduce graph terminology and some related notions.

1.2 Definitions

1.2.1 General Definitions

Given a set S, a *binary relation* R on S is a subset of the Cartesian product $S \times S$; in other words, it is a way of indicating which pairs of elements of S are related. A graph is a mathematical notation that is a useful representation of a binary relation.

Formally, a *graph* $G = (V, E)$ is a pair of sets: the *vertex set* V and the *edge set* E, where E is a set of pairs of elements of V. A graph is *simple* if $vv \notin E$ for all $v \in V$, and is *undirected* if $v_1v_2 \in E$ if and only if $v_2v_1 \in E$ for all $v_1, v_2 \in V$. For the remainder of this thesis, we assume that all graphs are undirected and simple.

Two vertices v_1 and v_2 are *adjacent* or *neighbours* if $v_1v_2 \in E$ and *nonadjacent* or *non-neighbours* if $v_1v_2 \notin E$. The set of neighbours of a vertex $v \in V$ in a graph $G = (V, E)$ is the *neighbourhood* of v, denoted $N_G(v)$ or, where clear from the context, $N(v)$. The *degree* of a vertex v in G is the number of vertices in its neighbourhood, denoted $deg_G(v)$. A *leaf* is a vertex in a graph with degree one.

A graph $H = (V_H, E_H)$ is a *subgraph* of a graph $G = (V_G, E_G)$ if $V_H \subseteq V_G$ and $E_H \subseteq E_G$. $H = (V_H, E_H)$ is an *induced* subgraph of $G = (V_G, E_G)$ if $V_H \subseteq V_G$ and $v_1v_2 \in E_H$ if and only if $v_1v_2 \in E_G$ for all $v_1, v_2 \in V_H$; we write $H = G[V_H]$.

A *path* of a graph is an ordered sequence $(v_0, v_1, ..., v_i)$ of pairwise distinct vertices, such that $v_j v_{j+1} \in E$, for $0 \leq j < i$; every other edge induced by the this vertex sequence is a *chord* of the path. A *cycle* of a graph is an ordered sequence $(v_0, v_1, ..., v_i)$ of pairwise distinct vertices, such that $v_j v_{j+1} \in E$ and $v_0 v_i \in E$, for $0 \leq j < i$; every other edge induced by the this vertex sequence is a *chord* of the cycle. We denote a chordless path P_n and a chordless cycle C_n, where n is the number of vertices in the path or cycle, respectively. The *length* of P_n and C_n is the number vertices n. Two vertices are *connected* if there exists a path between them. A graph $G = (V, E)$ is *connected* if every pair of vertices in V is connected. The distance between two connected vertices in G, denoted $d_G(v_1, v_2)$, is the length of the shortest path between v_1 and v_2 in G or, where clear from the context $d(v_1, v_2)$.

Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are *isomorphic*, written $G_1 \cong G_2$, if there exists a bijection f from V_1 to V_2 such that $v_1 v_2 \in E_1$ if and only if $f(v_1)f(v_2) \in E_2$. For a fixed graph H, a graph G is *H-free* if it does not contain any induced subgraph isomorphic to H.

A *forest* is an acyclic graph. A *tree* is a connected forest. For a discussion of the many useful properties of trees the reader is referred to [6, 41, 42].

Let A and B be two sets. The *intersection* of A and B, denoted $A \cap B$, is the set of elements contained in both A and B. The *union* of A and B, denoted $A \cup B$, is the set of elements contained in either A or B. The *set difference* of A from B, denoted $B \setminus A$, is the set of elements contained in B but not A.

1.2.2 Complexity

As the focus of this thesis is the development of efficient algorithms to solve various problems from computational biology, some complexity theory will be needed. We assume the reader is familiar with the concepts of algorithms, time complexity analysis, and NP-completeness. The texts [11, 18] both contain introductions to these concepts. All runtimes given in this thesis are with respect to the size of the input graph, namely the sum of the number vertices plus the number edges.

1.2.3 Chordal Graphs

A graph is *chordal* if every cycle of length four or more has a chord. Equivalently, a graph is chordal if it contains no induced cycle of length four or more.

Lemma 1.2.1. *[13] There exists a linear time algorithm to decide if a graph is chordal.*

One such algorithm based on lexicographical breadth first search (LexBFS)[37], a variant of the well studied breadth first search (BFS)[11] , searches as in BFS, but uses a lexicographical ordering of the vertices to choose the next vertex to search from. This algorithm exploits a structural property of chordal graphs called a simplicial vertex elimination ordering. A *simplicial vertex* is a vertex whose neighbourhood induces a clique. Dirac [13] and independently Lekkerkerker and

Boland [25] showed that all chordal graphs have a simplicial vertex. A *simplicial vertex elimination ordering* is a permutation of the vertices $(v_{s(0)}, v_{s(1)}, ..., v_{s(n-1)})$ such that $v_{s(i)}$ is simplicial in the subgraph graph induced by $\{v_{s(0)}, v_{s(1)}, ..., v_{s(i-1)}\}$ for $0 \leq i \leq (n-1)$. The algorithmic approach of simplicial vertex elimination ordering was introduced by Fulkerson and Gross who showed that a graph is chordal exactly when there exists a simplicial vertex elimination ordering [17]. The traversal order of the vertices by LexBFS is a simplicial ordering exactly when the graph is chordal [34].

For a graph $G = (V, E)$, $C \subset V$ is a *cutset* if the number of connected components of $G[V \setminus C]$ is greater than the number of connected components of G. A cutset C is *minimal* if no subset of C is a cutset. Dirac's theorem [13] states that a graph is chordal if and only if every minimal cutset in every induced subgraph is a clique.

1.2.4 Graph Powers and Graph Roots

For a graph $G = (V, E)$, the *kth power* of G is the graph $G^k = (V, E^k)$ such that

$$E^k = \{v_1v_2 \mid d(v_1, v_2) \leq k, v_1 \neq v_2\}.$$

A *kth root* of a graph G is a graph H such that $G = H^k$.

Computing the kth power of a graph $G = (V, E)$ can be done in $O(|V|^3)$ time using the Floyd-Warshall all-pairs shortest path algorithm [15, 40]. Conversely, finding the kth root of a graph has no known polynomial time algorithms; moreover, computing a square root of a graph is NP-complete [31]. If extra conditions are required of a root, polynomial time algorithms are known. For example, recognizing if a graph is the square of a tree and constructing such a root, if it exists, can be done in $O(|V|^3)$ time [29]. Similarly, recognizing if a graph is the kth power of a tree and constructing this tree can be done in $O(|V|^3)$ time [22].

1.2.5 Leaf-Labeled Trees and Steiner Points

For a tree, a vertex is *internal* if it is not a leaf. For a set S, a *leaf-labelled tree T corresponding to S* is a tree with an injective mapping from the leaves in T to the set S. Thus, a leaf-labelled tree T has three kinds of vertices or points: leaves which (all corresponding to vertices of S), internal points corresponding to vertices of S, and internal points which do not corresponding to vertices of S, called *Steiner points*.

Definition 1.2.1. *Given a graph $G = (V_G, E_G)$, a leaf-labelled tree $T = (V_T, E_T)$ corresponding to set V_G, and a positive integer k, define the following conditions of T:*

1. *the mapping from leaves in T to V_G is surjective,*

2. *every Steiner point in V_T has degree at least three,*

3. *G equals the subgraph of T^k induced by V_G.*

The first condition implies that the leaves of T are exactly the vertices of G, so all internal points of T are Steiner points. The second condition is based on the idea of an internal point representing a genetic split from a common ancestor into two or more descendants. The final condition requires that all adjacent units in G are within distance of k in T. We now define three problems which motivate our kth phylogenetic root construction method.

kTH PHYLOGENETIC ROOT PROBLEM (k-PRP) [27]

Instance: A graph $G = (V_G, E_G)$ and a positive integer k.

Query: Does a kth phylogenetic root tree T exist such that T is a leaf-labelled tree corresponding to set V_G and T satisfies conditions 1, 2 and 3 of Definition 1.2.1?

kTH LEAF ROOT PROBLEM (k-LRP) [32]

Instance: A graph $G = (V_G, E_G)$ and a positive integer k.

Query: Does a kth phylogenetic root tree T exist such that T is a leaf-labelled tree corresponding to set V_G and T satisfies conditions 1 and 3 of Definition 1.2.1?

kTH STEINER ROOT PROBLEM (k-SRP) [27]

Instance: A graph $G = (V_G, E_G)$ and a positive integer k.

Query: Does a kth phylogenetic root tree T exist such that T is a leaf-labelled tree corresponding to set V_G and T satisfies condition 3 of Definition 1.2.1?

Lin et al. showed that k-PRP has a linear time solution for $k \leq 4$ [27]. Chen et al. [8] demonstrated a linear time algorithm for $k \geq 2$ if T has bounded degree. Similarly, Nishimura et al. showed that k-LRP has a polynomial time solution for $k \leq 4$ [32]. In addition, Kennedy et al. showed that all strictly chordal graphs (defined in Chapter 2) have a k-leaf root tree for $k \geq 4$ [24]. Lin et al. showed that k-SRP is known to have a linear time solution for $k \leq 2$, [27]. Kennedy et al. showed that if the input graph is strictly chordal then 3-SRP has a linear time solution [24]. If $k \geq 5$, the complexity of both k-PRP and k-LRP, with respect to having a polynomial time algorithm or being in the class of NP-complete problems, is still an open question. Similarly, for k-SRP the complexity is still unknown for $k \geq 3$.

Lemma 1.2.2. *[27] A graph G has a kth phylogenetic root tree T, then G is chordal.*

Proof. It is known that the kth power of a tree is chordal for all positive integers k [29]. From the definition of chordal, every induced subgraph of a chordal graph is chordal. Therefore, the kth power of T is chordal and so the subgraph of T^k induced by V_G is also chordal. □

Since a kth root phylogenetic tree satisfies both Definitions 1.2.5 and 1.2.5, it follows that the preceding lemma holds for kth Steiner root trees as well as kth leaf root trees.

1.3 Overview

The following is a brief overview of the organization of this thesis. We first note that proofs of cited lemmas or theorems are given if the original proof has been change or altered in some fashion.

Proofs included from references [24, 23, 28] are shown if their proof is integral to the development of the final algorithm of this thesis. The majority of this thesis is original work.

Chapter 2: In Section 2.1, we introduce strictly chordal graphs, a subclass of chordal graphs for which structural properties allow efficient solutions to be developed for all three leaf-labeled root problems. We first develop the characterization that uses the notion of dual hypergraphs. Three other characterizations are presented which become the working definitions for the remainder of thesis. In Section 2.2.2 we introduce the class of tree chordal graphs and three equivalent characterizations. We show how to recognize both strictly chordal and tree chordal graphs in linear time.

Chapter 3: We describe the approach and method used for the construction of the 5th root phylogenetic tree along with some important preliminaries. In Section 3.1, we introduce a variation of k-SRP and show that it is equivalent the k-PRP. In Section 3.2, the basis for the reduction used to solve k-PRP for strictly chordal graph is presented – decomposition of a strictly chordal graph into a forest of tree chordal graphs. In Section 3.3, we overview the algorithm design for k-PRP.

Chapter 4: We consider the 5-PRP on the restricted class of tree chordal graphs. In Sections 4.1.1 – 4.1.3, we a solution for the 5-PRP where the input graph is tree chordal in three progressively less restrictive steps. In Section 4.2.2, we present a modification of the phylogenetic tree construction in Section 4.1.3 that will be used in the construction of k-PRP algorithm for strictly chordal graphs.

Chapter 5: We here discuss the main result of the thesis, the construction of 5-PRP for strictly chordal graphs, and the structural results that lead to its proof. In Section 5.1.1 we present several lemmas demonstrating the restrictive structure of maximal cliques containing three or more critical cliques. In Section 5.2, we progressively present the construction of the 5th phylogenetic root problem.

Chapter 6: We summarize the results presented in the previous chapters along with several open problems.

Chapter 2

Strictly Chordal Graphs

2.1 Preliminaries

2.1.1 Critical Clique Graphs

Let $G = (V, E)$ be a graph. A *clique* is a set of pairwise adjacent vertices. Denote a clique on k vertices as K_k. A clique is *maximal* if it is not properly contained in any other clique. A *homogeneous clique* is a clique S such that either $|S| = 1$ or for all $v_1, v_2 \in S$ and $w \in V \setminus S$, $v_1 w \in E$ if and only if $v_2 w \in E$. A *critical clique* is a homogeneous clique that is not a proper subset of any other homogeneous clique [27]. The *critical clique cardinality* of a maximal clique K, denoted *cccard*(K), is the number of critical cliques it contains. For convenience, we define a maximal clique to be *large* if it has critical clique cardinality three or more. The *size* of a critical clique C in a graph G is the number of vertices it contains. A critical clique is *internal* if it is contained in at least two maximal cliques and *external* otherwise. Figure 2.1 illustrates these concepts.

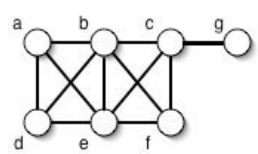


Figure 2.1: The maximal cliques in the above graph are {a,b,d,e}, {b,c,e,f}, and {c,g}. The critical cliques in the above graph are {a,d}, {b,e}, {c}, {f}, and {g}. The cardinality of maximal clique {a,b,d,e} is 2. The critical clique {a,d} is external, whereas the critical clique {b,e} is internal.

Define a *partition* of a nonempty set S as a set of nonempty sets $\{S_1, S_2, .., S_p\}$ such that $S_i \cap S_j = \emptyset$, for each $i \neq j$, and $S_1 \cup S_2 \cup ... \cup S_p = S$. The set of maximal cliques of a graph does not form a vertex partition, as maximal cliques can overlap. On the other hand, the set of critical cliques of graph does form a partition of the vertex set.

Lemma 2.1.1. *[27] Let $G = (V, E)$ be a graph and S =$\{S_1, S_2, ..., S_p\}$ be the set of critical cliques in G. Then S forms a partition of V.*

Two critical cliques are *adjacent* if they are part of a common maximal clique. Define the *critical clique graph* of a graph G, denoted $CC(G)$, where the vertex set is the set of all critical cliques in G and two vertices are adjacent if and only if the critical cliques they represent are adjacent in G [27]. By the definition of adjacency and Lemma 2.1.1, it follows that $CC(G)$ is well-defined for all G.

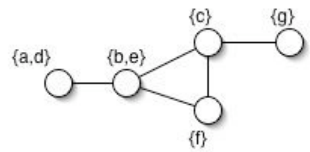


Figure 2.2: The critical clique graph for the graph in Figure 2.1.

Lemma 2.1.2. *[27] Let G be a graph. Then $CC(G) = CC(CC(G))$.*

Lemma 2.1.3. *[27] Let G be a chordal graph. Then there exists an $O(|V| + |E|)$ time algorithm to construct the critical clique graph $CC(G)$.*

2.1.2 Hypertrees

Hypergraphs, or set systems, are natural extensions of graphs where the edge set is generalized from a vertex pair to a vertex subset. Specifically, a *hypergraph* $\mathcal{H} = (V, \mathcal{E})$ consists of a vertex set V and a hyperedge set $\mathcal{E} = \{e_1, e_2, \ldots, e_m\}$, where $e_i \subseteq V$ for $1 \leq i \leq m$ [3]. A hypergraph is *connected* if for every pair of vertices u and v there is a sequence of hyperedges ($e_{\alpha(1)}$, $e_{\alpha(2)}$, . . ., $e_{\alpha(\ell)}$) such that $u \in e_{\alpha(1)}$, $v \in e_{\alpha(\ell)}$, and $e_{\alpha(j)} \cap e_{\alpha(j+1)} \neq \emptyset$ for $1 \leq j < \ell$.

Define the *clique hypergraph* of $G = (V, E)$ as the hypergraph $\mathcal{H}(G) = (V, \mathcal{E})$ where $\mathcal{E}$ is the set of maximal cliques of G.

Let $\mathcal{H} = (V, \mathcal{E})$ be a hypergraph. $\mathcal{H}$ is a *hypertree* if there exists a tree T with vertex set V such that every hyperedge in $\mathcal{E}$ induces a subtree in T [3]. A graph G is *dually chordal* if its clique hypergraph is a hypertree [5].

$\mathcal{H}$ is a *dual hypertree* if there exists a tree T with vertex set $\mathcal{E}$ such that, for each vertex of $\mathcal{H}$ the set of hyperedges containing that vertex form a subtree of T [6].

Let $\mathcal{H}(V, \mathcal{E})$ be a hypergraph. A hyperedge $e_t \in \mathcal{E}$ is a *twig* if for $\mathcal{R} = \cup_{e \in \mathcal{E} - e_t} e$ either

$$e_t \cap \mathcal{R} = \emptyset,$$

or, for some $e_b \in \mathcal{E} - e_t$

$$e_t \cap \mathcal{R} = e_t \cap e_b.$$

e_b is a *branch* for the twig e_t [21].

A *twig elimination ordering of a hypergraph* [21] is a total ordering of the hyperedges $(e_{\alpha(1)}, e_{\alpha(2)}, \ldots, e_{\alpha(m)})$ such that for $1 \leq i \leq m$ e_i is a twig in the sub-hypergraph $\mathcal{H}_i = (V_i, \mathcal{E}_i)$, where $V_i = e_{\alpha(1)} \cup e_{\alpha(2)} \cup \ldots \cup e_{\alpha(i)}$ and $\mathcal{E}_i = \{e_{\alpha(1)}, e_{\alpha(2)}, \ldots, e_{\alpha(i)}\}$.

Theorem 2.1.4. *[7, 19, 39] (See [20]) Let $G = (V, E)$ be a graph. Then the following are equivalent:*

- *G is chordal.*
- *G is the intersection graph of subtrees of a tree.*
- *$\mathcal{H}(G)$ is a dual hypertree.*

Notice that Theorem 2.1.4 can be easily adapted to show the following lemma.

Lemma 2.1.5. *A graph G is chordal if and only if $\mathcal{H}(G)$ has a twig elimination ordering.*

Proof. We will prove the forward direction by induction on the number of vertices of G. A graph with a single vertex is chordal and has a clique hypergraph with a single twig. Assume that all chordal graphs with n vertices have a valid twig elimination ordering; let G_{n+1} be a chordal graph on $n+1$ vertices. As G_{n+1} is chordal, let $(v_{s(1)}, v_{s(2)}, ..., v_{s(n+1)})$ be a simplicial vertex elimination ordering on the vertex set. Let G_n be the graph formed by removing vertex $v_{s(n+1)}$ and all edges incident with it. G_n is chordal and has n vertices; thus by the inductive hypothesis, the clique hypergraph of G has a twig elimination ordering $K_1, K_2, ..., K_p$. Let S be the set of maximal cliques such that $N_{G_{n+1}}(v_{s(n+1)})$ is a subset of maximal cliques in S. If, for a maximal clique $K \in S$, $N_{G_{n+1}}(v_{s(n+1)}) = K$ then $K \cup v_{s(n+1)}$ forms a new larger maximal clique and the same ordering of hyperedges is a twig elimination ordering. If $N_{G_{n+1}}(v_{s(n+1)})$ is a proper subset of every maximal clique in S then let K_i be the first maximal clique in the twig elimination ordering which is also in S. Place the new maximal clique, $K' = v_{s(n+1)} \cup N_{G_{n+1}}(v_{s(n+1)})$ immediately before K_i in the twig elimination ordering. It follows that K' is a twig with K_i as its branch, as $K' \cap K_i$ intersection is the intersection of K' with the graph. Therefore, G_{n+1} has twig elimination ordering. Thus, G chordal implies the clique hypergraph of G has a twig elimination ordering.

We will show the contrapositive of the backwards direction. Assume G is not chordal. Therefore, G has a chordless cycle, $C_i = (v_1, v_2, ..., v_i)$, where $i \geq 4$. Note that each pairwise adjacent set of vertices in the cycle is contained in a maximal clique which is a hyperedge in the clique hypergraph. Assume that there exists a valid twig elimination ordering and let e_j be the hyperedge corresponding to a pair a vertices $v_j, v_{j+1(mod\ i)}$ such that e_j is the first hyperedge in the ordering containing both v_j and $v_{j+1(mod\ i)}$. Such a hyperedge must exist as all hyperedges will be eventually removed. Therefore, e_j is the first hyperedge removed as a twig that will remove an edge in the cycle. Let $e_{j-1(mod\ i)}$ be a the hyperedge containing $v_{j-1(mod\ i)}$ and v_j and $e_{j+1(mod\ i)}$ be a hyperedge containing $v_{j+1(mod\ i)}$ and $v_{j+2(mod\ i)}$. Both edges will still be in the graph as we assumed that e_j was the first hyperedge corresponding to an edge in the cycle. Also, as $v_{j-1(mod\ i)}$ and $v_{j+1(mod\ i)}$ are non-adjacent, $v_{j+1(mod\ i)} \notin e_{j-1(mod\ i)}$. Similarly, v_j and $v_{j+1(mod\ i)}$ are non-adjacent so $v_j \notin e_{j+1(mod\ i)}$. Therefore, e_j can not be a twig as no other hyperedge contains both v_j and v_{j+1}, a contradiction to this being a valid twig elimination ordering. Therefore, as any twig elimination ordering will have such an edge, no valid ordering will exist. Thus, if G is chordal the clique hypergraph for G has a valid twig elimination ordering. $\square$

Corollary 2.1.6. *Let $\mathcal{H}$ be a hypergraph. Then $\mathcal{H}(G)$ is a dual hypertree if and only if $\mathcal{H}(G)$ has a twig elimination ordering.*

2.1.3 Moplexes

As the twig elimination ordering suggests, chordal graphs can be viewed as a generalization of trees. A further generalization of this ordering leads to a moplex elimination ordering [4].

Let $G = (V, E)$ be a graph. A *moplex* is a critical clique whose neighbourhood is a minimal cutset of G. A moplex is *simplicial* if all vertices in the moplex are simplicial. A *simplicial moplex elimination ordering* is a sequence $M_1, M_2, ..., M_\ell$, such that M_i is a simplicial moplex in the $G(M_1 \cap M_2 \cap ... \cap M_i)$, for $1 \leq i \leq \ell$.

Lemma 2.1.7. *Let $G = (V, E)$ be a chordal graph. If M is a simplicial moplex in G, then $e = M \cup N(M)$ are the vertices of a hyperedge in the clique hypergraph $\mathcal{H}(G)$. Moreover, e is a twig in $\mathcal{H}(G)$.*

Proof. Let $G = (V, E)$ be a chordal graph and M be a simplicial moplex in G. By Dirac's theorem [13], every minimal cutset is a clique in a chordal graph. As $N(M)$ is a minimal cutset and as M is a critical clique, it follows that $e = M \cup N(M)$ is a clique in G. As M is made of simplicial vertices, all vertices in M are adjacent only to other vertices in M and vertices in $N(M)$. As such, no other vertex exists that is adjacent to every vertex in $N(M) \cup M$, therefore, it must correspond to a maximal clique, and thus, a hyperedge in $\mathcal{H}(G)$.

To see that e is a twig, we must find a branch. We claim that $N(M)$ must be contained in another maximal clique in G. Assume that this not the case, and there exists no vertex adjacent to all vertices in $N(M)$. There must exist at least one vertex in $v \in (V \setminus M)$ that is adjacent to a vertex in $N(M)$, as otherwise, it is not a minimal cutset. It follows that the vertices in the cutset adjacent to v form a smaller cutset, contradicting the minimality of $N(M)$. Therefore, at least one vertex must be adjacent to all of $N(M)$, and therefore, the maximal clique containing this vertex and $N(M)$ forms a branch for M in $\mathcal{H}(G)$. It follows that a moplex in a chordal graph is a twig in the clique hypergraph of G. $\square$

2.2 Strictly Chordal Graphs

2.2.1 Chararcterizations

Let $\mathcal{H} = (V, \mathcal{E})$ be a hypergraph. Let $\mathcal{E}' = \{e_{\alpha(1)}, e_{\alpha(2)}, \ldots, e_{\alpha(\ell)}\}$ with $\ell \geq 2$, be a subset of $\mathcal{E}$ with non-empty intersection, namely $I(\mathcal{E}') = \cap_{j=1}^{\ell} e_{\alpha(j)} \neq \emptyset$. $\mathcal{E}'$ is *intersection maximal* if every hyperedge which intersects $I(\mathcal{E}')$ is contained in $\mathcal{E}'$. $I(\mathcal{E}')$ is a *strict* intersection if $\mathcal{E}'$ is intersection maximal, and if, for every pair of hyperedges $e', e'' \in \mathcal{E}'$, $e' \cap e'' = I$. A hypergraph is *strict* if for every subset $\mathcal{E}'$ of $\mathcal{E}$ such that $\mathcal{E}'$ is intersection maximal, $I(\mathcal{E}')$ is strict.

A *dart* graph is any graph isomorphic to the graph on vertices a, b, c, d, e with edges (a, b), (b, c), (b, d), (b, e), (c, e), and (d, e) (see Figure 2.2.1). Dart-free graphs have been studied in the context of perfect graphs; a graph is *perfect* if its chromatic number[1] is equal to the size of a largest clique for all of its induced subgraphs. Sun [38] showed that the Perfect Graph Theorem[2] (then conjecture) holds for this class of graphs. Chvatal et al. [10] showed that perfect dart-free graphs can be recognized in polynomial time.

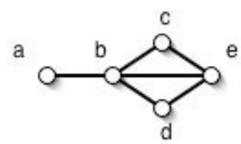


Figure 2.3: A labelled dart graph.

A *gem* graph is any graph isomorphic to graph on vertices a, b, c, d, e with edges (a, b), (a, c), (b, c), (b, d), (b, e), (c, e), and (d, e) (see Figure 2.2.1). A *wheel* graph, denoted W_n, is any graph isomorphic to C_n with an additional vertex adjacent to each vertex in the cycle C_n (see Figure 2.2.1).

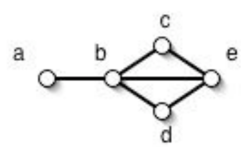


Figure 2.4: A labelled gem graph.

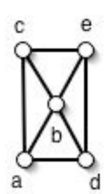


Figure 2.5: A labelled 4-wheel graph, W_4.

Lemma 2.2.1. *Let G be a graph. $\mathcal{H}(G)$ has all intersections strict if and only if G is dart-free, gem-free and W_4-free.*

[1]The chromatic number $\chi(G)$ of graph G is the minimum number of labels needed to label the vertices of a graph such that adjacent vertices receive different labels.

[2]The Perfect Graph Theorem states that a graph is perfect if and only if no induced subgraph contains an odd cycle of length 5 or more or the complement of an odd cycle of length 5 or more. It was conjectured in 1960 by Claude Berge [2] and proven by Maria Chudnovsky, Paul Seymour, Neil Robertson and Robin Thomas in 2002 [9].

Proof. Given a graph G such that $\mathcal{H}(G)$ all strict intersections, assume that the vertices a, b, c, d, e induce a dart in G. Let e_{ab} be the hyperedge containing vertices ab, let e_{bce} be the hyperedge containing bce, and let e_{bde} be the hyperedge containing bde. But the maximal clique ab intersects only part of the intersection of maximal cliques bce and bde, so the intersection of e_{ab}, e_{bce}, e_{bde} is not strict, a contradiction. Similarly, if a, b, c, d, e induce a gem, we have the same hyperedge set with, in addition, hyperedge e_{ab} now containing the vertex c, denote e_{abc}. The edge e_{abc} intersects only part of the intersection of e_{bce} and e_{bde}, therefore, not a strict intersection. If a, b, c, d, e induce a W_4, we the same hyperedge set with the addition of hyperedge e_{abd}. The same non-strict intersection as for the gem graph will occur.

Conversely, assume that $\mathcal{H}(G)$ contains a non-strict intersection I. We will show that G contains either a dart, gem or W_4 graph as an induced subgraph. Therefore, take a set of hyperedges $\{e_1, e_2,$ ..., $e_\ell\}$ which are a part of the non-strict intersection such that $|e_1 \cap e_2| > 1$ and e_3 intersects only a proper subset of $e_1 \cap e_2$. A non-strict intersection must involve at least three hyperedges. Any intersection that involves exactly two hyperedges, the intersection is a clique in G, all vertices in the intersection share the same neighbourhood outside of the clique in G, and as they are an intersection no other vertex exists satisfying the first two properties, thus they form a critical clique in G. As an intersection which is a critical clique in G is a strict intersection, we have that $\ell \geq 3$. Also, at least two hyperedges must intersect by two or more vertices, as if all intersect by a single vertex again it is a critical clique. And finally, at least one hyperedge must have an intersection which is a proper superset of I, as if all pair-wise intersections are exactly I then the intersection is a critical clique. Define the following vertices; a vertex a from $e_3 \setminus (e_1 \cup e_2)$, a vertex b from $e_1 \cap I$, a vertex e from $I \setminus e_3$, a vertex c from $e_1 \setminus I$, a vertex d from $e_2 \setminus I$. We show the vertices a, b, c, d, e either induce a dart, a gem, or a W_4 graph in G. If c is contained only in e_1 and d is contained only in e_2, then we have an induced dart. If exactly one of c and d, without loss of generality assume c, is contained in e_3 and there exists a hyperedge containing the vertices a, b, d, then we have an induced W_4. If no such hyperedge exists, we have an induced gem. □

Let G be a graph and $\mathcal{H}(G)$ its clique hypergraph. $\mathcal{H}(G)$ is a *strict dual hypertree* if $\mathcal{H}(G)$ is strict and a dual hypertree. G is *strictly chordal* if $\mathcal{H}(G)$ is a strict dual hypertree. By Lemma 2.1.5, it follows that strictly chordal graphs live up to their name and are chordal. The following is another characterization of strictly chordal graphs which follows as a corollary of Lemmas 2.2.1 and 2.1.5.

Corollary 2.2.2. *Let G be a graph. G is strictly chordal if and only if it is dart-free, gem-free and chordal.*

Lemma 2.2.3. *Let G be a graph. G is strictly chordal if and only if G is chordal and every set of maximal cliques has intersection of either exactly one critical clique or* $\emptyset$.

Proof. A strictly chordal graph G is chordal and the clique hypergraph of G is a strict dual hypertree. Assume there exist two maximal cliques, K_1 and K_2, in $CC(G)$ with two or more vertices in common. Each of these common vertices must have a different neighbourhood in both $CC(G)$ and G as they represent different critical cliques. Therefore in G there must exist a third maximal clique which is adjacent to only part of the intersection of K_1 and K_2; this intersection is not strict. This contradicts the clique hypergraph of G being a strict dual hypertree.

Let G be a chordal graph such that for every subset of maximal cliques in $CC(G)$, the intersection is either $\emptyset$ or a single vertex in $CC(G)$. As shown above, since G is chordal, the clique hypergraph $\mathcal{H}(G)$ must be a dual hypertree. In $\mathcal{H}(G)$, take any set of hyperedges $\{e_1, e_2, ..., e_i\}$ having a nonempty intersection, say vertex v. Notice in $CC(G)$ the critical clique containing v is represented by a single vertex, therefore, no vertex not in this critical clique is contained in more than one of these hyperedges as two maximal cliques would share two vertices in common in $CC(G)$. Therefore the hyperedges form a strict intersection. As this argument holds for all intersections, all intersections are strict in the clique hypergraph. Therefore, as G's clique hypergraph is a strict dual hypertree, so G is strictly chordal. $\square$

Lemma 2.2.4. *[24] For a chordal graph G, the clique hypergraph $\mathcal{H}(G)$ is a strict dual hypertree if and only if for every (not necessarily induced) cycle the vertices induce a clique in the critical clique graph $CC(G)$.*

Proof. Given G with $CC(G)$ satisfying that each cycle is a clique, it follows G is chordal as $CC(G)$ largest induced cycle has three vertices and replacing vertices of $CC(G)$ with their corresponding critical cliques will maintain this property. Therefore from Lemma 2.1.5 it follows that the clique hypergraph of G is a dual hypertree. $CC(G)$ must be dart-free and gem-free as both darts and gems contain non-clique cycle. It follows that G must also be dart-free and gem-free as replacing vertices of $CC(G)$ with their corresponding critical cliques again maintains this property. Therefore, G is strictly chordal.

Assume that $\mathcal{H}(G)$ is a strict dual hypertree, as G is chordal, so must be $CC(G)$, therefore all cycles of length four or more will have a chord. For contradiction assume that there exists a simple cycle in $CC(G)$ which has at least four nodes and does not form a clique. Pick the shortest of all these cycles, $(C_0, C_1, \ldots, C_{\ell-1})$. There must exist two nodes in $CC(G)$ such that C_i and $C_{i+2(mod\ \ell)}$ are non-adjacent, as otherwise, if $\ell = 4, 5$ then the cycle forms a clique and if $\ell \geq 6$ the cycle is not minimal. Therefore, without loss of generality, assume C_0 and C_2 are not adjacent. Let K_0 be a maximal clique in G that includes C_0 and C_1, and K_1 be a maximal clique including C_1 and C_2.

Let $I = K_0 \cap K_1$; note by the definition of a strict dual hypertree $I = C_1$ which implies $C_i \cap I = \emptyset$ for $i \neq 1$. Let C_i be the last critical clique such that $C_{i+1} \not\subseteq K_1$. Let K_2 be a maximal clique including C_i and $C_{i+1(mod\ \ell)}$. The intersection $K_1 \cap K_2$ includes C_i and does not intersect $K_0 \cap K_1$. We continue to build this chain of maximal cliques until we find a $K_{\ell'-1}$ that intersects with K_0, leaving a sequence of at least three maximal cliques $K_0, K_1, \ldots, K_{\ell'-1}$ such that

$K_i \cap K_{i+1(mod\ \ell')}$ are the only non-empty pairwise intersections. Suppose without loss of generality that $K_{\ell'-1}$ is the first maximal clique that appears in a twig elimination ordering $(K_0, K_1, \ldots, K_m)$. Then we will find no branch for $K_{\ell'-1}$ since $K_{\ell'-1}$ intersects at least two other maximal cliques in the sub-ordering $(K_0, K_1, \ldots, K_{\ell'-1})$. This contradicts the assumption that the clique hypergraph $\mathcal{H}(G)$ is a strict dual hypertree. □

We have previously presented a forbidden induced subgraph characterization of strictly chordal graphs, namely that strictly chordal graphs do not contain any induced subgraph isomorphic to a dart, gem, or C_i for $i \geq 4$. It is also interesting to give a generation construction, such a construction will show exactly how strictly chordal graphs can be generated from small graphs. Hence we define the following operation. Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be graphs and let $c_1 \subseteq V_1$ and $c_2 \subseteq V_2$, where c_1, c_2 are cliques. The *clique join* of the pairs (G_1, c_1) and (G_2, c_2), denoted $(G_1, c_1) \star (G_2, c_2)$, is the graph $G = (V(G), E(G))$, such that

$$V(G) = V_1 \cup V_2,$$

$$E(G) = E_1 \cup E_2 \cup (c_1 \times c_2).$$

Lemma 2.2.5. *Let $G = (V, E)$ be a graph.*

1. *If $|V| = 1$, then G is strictly chordal.*

2. *If $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are vertex disjoint and strictly chordal graphs, then $G = (V_1 \cup V_2, E_1 \cup E_2)$ is strictly chordal.*

3. *If $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are vertex disjoint and strictly chordal and there are subsets $c_1 \subseteq V_1$ and $c_2 \subseteq V_2$ such that each is either a critical clique or is a clique contained in exactly one maximal clique, then $G = (G_1, c_1) \star (G_2, c_2)$ is strictly chordal.*

4. *All strictly chordal graphs can be generated using rules $1-3$.*

Proof. Trivially, a single vertex graph is a strictly chordal graph. In the second case, as no edges are added, the new graph is still strictly chordal.

For the third case, the new graph G will be chordal, as the clique join adds the complete set of edges between the vertices of c_1 and c_2, thus no cycle of length four or more can be created. Notice in the clique hypergraph $\mathcal{H}(G)$ that $c_1 \cup c_2$ forms a maximal clique and, thus, a hyperedge e in $\mathcal{H}(G)$. We show that this maximal clique either contains exactly one critical clique $c_1 \cup c_2$ or exactly two critical cliques c_1 and c_2. If c_1 and c_2 were both maximal cliques, then the clique join will be a larger maximal clique. Otherwise, c_1 will be a critical cliques in G as its new neighbourhood will be $N_{G_1}(c_1) \cup c_2$, similarly $N_G(c_2) = N_{G_1}(c_2) \cup c_1$. Therefore, as the hyperedge e has only strict intersections with $G(V_1)$ and $G(V_2)$ and as all intersections in G_1 and G_2 are strict, it follows that G is strictly chordal.

To see that there are no further strictly chordal graphs, we show that given any strictly chordal graph G, there exists a sequence of the above steps to produce G. Without loss of generality assume G is connected, as if we can produce a sequence for each connected component of G then we can apply the second rule to produce the whole G.

Let $(e_{\alpha(1)}, e_{\alpha(2)}, \ldots, e_{\alpha(m)})$ be a twig elimination ordering for $\mathcal{H}(G)$. We proceed by adding hyperedges from $e_{\alpha(1)}$ to $e_{\alpha(m)}$ creating graph $\mathcal{H}(G_i) = (V_i, \mathcal{E}_i)$, where $V_i = (e_{\alpha(1)} \cap e_{\alpha(2)} \cap \ldots \cap e_{\alpha(i)})$ and $\mathcal{E}_i = (e_{\alpha(1)}, e_{\alpha(2)}, \ldots, e_{\alpha(i)})$, for $1 \leq i \leq m$.

For a $e_{\alpha(i)}$, let $e' = e_{\alpha(i)} \cap V_i$ and $e'' = e_{\alpha(i)} \setminus e'$. Notice that both e' and e'' induce critical cliques in $\mathcal{H}(G)$. Trivially, e' can be constructed using single vertices and rule 3. Then $G_{i+1} = (G_i, e') \star (e'', e'')$. Completing this for each twig in the twig elimination ordering will construct G.

□

2.2.2 Tree Chordal Graphs

We now consider a subclass of strictly chordal graphs that will be used in the solution of 5-PRP. A graph G is *tree chordal* if $CC(G)$ is a tree [28].

Lemma 2.2.6. *A graph G is a tree chordal graph if and only if it is chordal and every maximal clique contains at most two critical cliques.*

Proof. Let G be a graph such that $CC(G)$ is a tree; we will show that G is chordal and every maximal clique contains at most two critical cliques. As $CC(G)$ collapses critical cliques in to single vertices, for any chordless path that exists in G a path of the same length can be found in $CC(G)$ by replacing vertices in the path with the critical clique they are part of. Any chordless path will contain at most one vertex from a critical clique, as the path is chordless. Therefore, if G is not chordal then we can find a chordal cycle in both G and $CC(G)$ a contradiction to $CC(G)$ being a tree; therefore, G is chordal. As maximal cliques in $CC(G)$ correspond directly to maximal cliques in G, $CC(G)$ being a tree implies that the largest clique is size two, implying that each maximal clique in G contains at most two critical cliques.

Let G be a chordal graph such that every maximal clique contains at most two critical cliques; we will show that $CC(G)$ is a tree. As every maximal clique contains at most two critical cliques, $CC(G)$ contains no clique larger than two vertices. Therefore, to show $CC(G)$ is a tree we need only show it does not contain any induced cycles of length four or more. For any induced cycle in $CC(G)$ a corresponding cycle can be found in G, by replacing a node in the cycle corresponding to a critical clique with any of the vertices in the critical clique. Therefore, as G is chordal we can not find any cycle of length four or more, and we will, therefore, not be able to find any such cycle in $CC(G)$. Therefore $CC(G)$ is a tree. □

Lemma 2.2.7. *A graph G is a tree chordal graph if and only if there exists a tree, T, such that G can be created by replacing each vertex, $v_i \in V(T)$ by a clique K_i and replacing each edge*

$(v_i, v_j) \in E(T)$ *by* $(K_i \times K_j)$.

Proof. Let G be a tree chordal graph. Let $T = CC(G)$. The lemma describes the reverse process of creating a $CC(G)$ from a graph G. Both directions follow easily from this observation. □

2.2.3 Recognition

Theorem 2.2.8. *[24] There exists a linear time algorithm for recognizing whether or not a graph is strictly chordal, and if so, returning its critical clique graph.*

Proof. We can decide if a graph is chordal in linear time, by Lemma 1.2.1; moreover, we can build the critical clique graph in linear time as well, by Lemma 2.1.3. By Lemma 2.2.4, it suffices to check that every simple cycle in $CC(GG)$ is a clique. The breadth first search (BFS), representing the order of edges searched, is known to have two types of edges: tree edges and cross edges [11]. A tree edge corresponds to a new vertex encountered while traversing the graph, whereas a cross edge corresponds to already encountered vertex. Necessarily, the cross edge set and the tree edge set are disjoint and cover the whole edge set of the graph. A cross edge represents two connected nodes which share a common ancestor that is not a parent; as such, if the nodes are not siblings in the tree, a cycle of length at least four exists in the graph which is clearly not a clique as they are not adjacent to both parents. To check for non-induced cycles, check the set of all child nodes of any fixed node in the BFS tree, the graph induced must be a collection of disjoint cliques, otherwise there exists a non-induced cycle whose nodes do not form a clique. BFS runtime is linear time in the size $CC(G)$ and it takes linear time to check the child node lists; as the number of edges and vertices of $CC(G)$ is bounded above by the number of edges and vertices in G, it follows that BFS is linear with respect to the size of G. □

Corollary 2.2.9. *There exists a linear time algorithm for recognizing whether or not a graph is tree chordal, and if so, returnin its critical clique graph* $CC(G)$.

Proof. Build $CC(G)$ using Lemma 2.1.3 in linear time. Using BFS to check if $CC(G)$ is acyclic and return yes if so. □

Chapter 3

Root Construction Methods

This chapter explains the approach we use in our algorithm for the construction of a 5th phylogenetic root tree. We introduce a variation of k-SRP that we will show is equivalent to k-PRP. We then present our algorithm design for the k-PRP serving as a guideline for the chapters to follow.

3.1 S-restricted kth Steiner Root Trees

An *S-restricted kth Steiner root tree* T for a set of critical cliques S of a graph G is a kth Steiner root tree T such that T has no degree 2 Steiner points and the representatives of critical cliques in S are internal in T. For each critical clique c, there exists a set of vertices $\{r_1, r_2, ..., r_n\}$ in an S-restricted kth Steiner root tree that correspond to c; each r_i represents a set of vertices in c. We denote each element of such a vertex r_i as a *representative of c*. The *size* of r_i as the number of vertices that map to it from c. No vertex in a critical clique has more than one representative, and every vertex in a critical clique corresponds to one of its representatives.

Lemma 3.1.1. *[23] Let G be a graph and let S be the set of all critical cliques of size 1 in G. Then G has an S-restricted (k-2)th Steiner root tree T if and only if G has a kth phylogenetic root tree.*

Proof. Assume that G has an S-restricted $(k-2)$th Steiner root tree T; we construct a kth phylogenetic root tree T'. Replace each representative of a critical clique with a Steiner point and set adjacent to this point each of the vertices represented. As T was a valid $(k-2)$th Steiner root tree T, it follows that adjacent representatives are at a distance of at most $k-2$. The construction extends all paths between every vertex by exactly two, therefore, all adjacent vertices are at a distance of at most k in T'. Similarly, nonadjacent representatives are at a distance of at least $k-1$ in T; it follows that they will be at distance at least $k+1$. Steiner points in T maintain their degree in T' and therefore still at least 3. Representatives that were leaves in T had size at least 2 therefore, the created Steiner point has degree at least 3. Representatives that were internal had degree at least 2 will now have degree at least 3 as a Steiner point as they represent at least one vertex. Therefore, Steiner points all have degree at least 3. Thus, as all vertices are represented by leaves it follows that T' is a valid 5th phylogeny root tree.

Assume that G has a kth phylogenetic tree T; we construct an S-restricted $(k-2)$th Steiner root tree T'. To construct such a tree, for each Steiner point in Twe record the number of leaves it is adjacent to and then remove all leaves in the T to produce T'. Steiner points in T that were not adjacent to any leaves will remain as Steiner points in T'; notice that they will still have degree of three or more. The remaining Steiner points now correspond to vertices in the input graph; let the set S be the internal representatives in the graph of size 1. T' is a valid S-restricted kth Steiner root tree with S corresponding to all size 1 vertices; these vertices were are internal as T was a valid 5th phylogenetic root tree. $\square$

Using the preceding lemma, our algorithm for 5-PRP searches for an S-restricted 3rd Steiner root tree.

3.2 Decomposition of Strictly Chordal Graphs

Our algorithm for 5-PRP for strictly chordal graphs uses the algorithm for 5-PRP for tree chordal graphs and the following observation. Observe that we can produce a set of tree chordal graphs $\mathcal{T}$ from a strictly chordal graph in the following way. Using the critical clique graph $CC(G)$, remove the edges from large maximal cliques to produce $CC(\mathcal{T})$, which is a set of trees. To create $\mathcal{T}$, re-substitute each critical clique in for the node that it represents in $CC(\mathcal{T})$. Figure 3.1 gives an example.

(a) G.

(b) Forest $\mathcal{T}$.

Figure 3.1: An example decomposition from a strictly chordal graph to a forest of tree chordal graphs.

It follows from the input graph being strictly chordal that each node is part of exactly one tree chordal graph.

3.3 kth Phylogenetic Root Algorithm

The following is a brief overview of the strategy we employ to produce a 5th phylogeny root tree from a graph G. Starting with G, check if G is strictly chordal, and if yes, build the critical clique graph $CC(G)$.

Using $CC(G)$, we produce the set of tree chordal graphs $\mathcal{T}$ by the decomposition of Section 3.2. For each tree chordal graph $T_i \in \mathcal{T}$, we modify the critical cliques contained in large maximal cliques in G to produce T_i'. We justify this modification in Section 4.2.2.

Let S be the size 1 nodes in T_i'. We search for an S-restricted 3rd Steiner root tree T_i'. If no S-restricted 3rd Steiner root tree exists, then no phylogeny tree exists by Lemma 4.2.1. If for all $T_i' \in \mathcal{T}$, there exists a Steiner tree S_i we then continue to consider the edges removed from the large maximal cliques.

We combine each 3rd Steiner root tree S_i until either we come to a contradiction, or we have a valid S-restricted 3rd Steiner root tree where S is the set of critical cliques of size 1 in G. If we find such a Steiner tree, then by Lemma 3.1.1 we are always able to produce a corresponding 5th phylogeny root tree. Figure 3.2 shows a simplified flow chart of the steps of our algorithm.

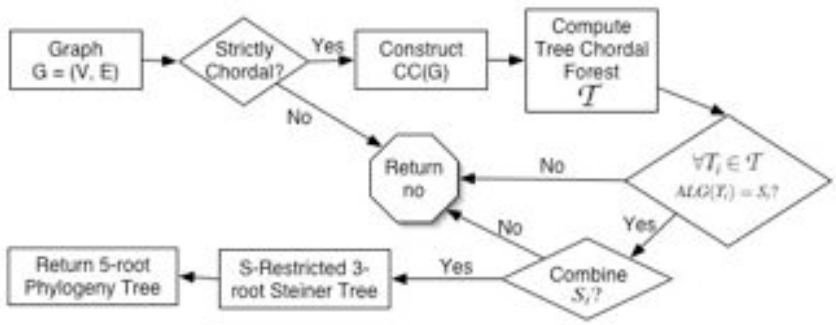


Figure 3.2: A flow chart of our 5th phylogeny root tree construction algorithm. $ALG(T_i)$ is our algorithm for S-restricted 3-SRP, as described in Section 4.2.2.

Chapter 4

Root Construction for Tree Chordal Graphs

We now consider the 5-PRP construction for tree chordal graphs in general and then we consider the construction as an intermediate step in the algorithm to solve 5-PRP for strictly chordal graphs. For this chapter[1], we primarily consider the S-restricted 3rd Steiner root tree construction problem. We remind the reader that by Lemma 3.1.1, when the set S contains exactly all size 1 critical cliques in the input graph G, the problem of constructing an S-restricted 3-SRP tree is equivalent to constructing a 5-PRP tree.

Note that every maximal clique in a tree chordal graph G contains exactly two critical cliques. We assume for this chapter that the given graph G is tree chordal and its critical clique graph $CC(G)$ has been constructed.

4.1 General Construction

We assume, for this section, that every graph contains at least three critical cliques, as such, $CC(G)$ will have at least three nodes. We also assume, for this section, that S corresponds to the size 1 critical cliques in G.

4.1.1 Structural Restriction 1

Lemma 4.1.1. *Let G be a tree chordal graph with an S-restricted 3rd Steiner root T. Let u denote a leaf node in $CC(G)$. Then removing the representatives for u does not disconnect T.*

Proof. Let v denote the internal node adjacent to u in $CC(G)$. Note that all representatives for v must be within distance 3 to $r(u)$ but no representative for a critical clique other than u and v can be within distance 3 to $r(u)$. Assume, for contradiction that removing the representatives for u from Steiner root T disconnects T. It follows that there must be some representative $r(u)$ for u that lies on the path P connecting representatives of two other critical cliques. It follows that one of these

[1] A version of this chapter has been submitted for publication. Lin, Kennedy, Kong and Yan 2005. Discrete Applied Mathematics [28]

critical cliques is not v and its representatives must be at a distance of at least 4 from $r(u)$, otherwise, they are representatives for v. Therefore, this critical clique is nonadjacent, a contradiction to the existence of such a path. □

Corollary 4.1.2. *Let G be a tree chordal graph with an S-restricted 3rd Steiner root T. Then there exist no size 1 leaf nodes in $CC(G)$.*

Proof. Let u be a size 1 leaf node in $CC(G)$. u will have exactly one representative in any 3rd Steiner root tree T. By Lemma 4.1.1, if u was internal removing it would disconnect T, implying that u has two neighbours in $CC(G)$, a contradiction. Thus, u is a leaf in any 3rd Steiner root tree T, specifically in any S-restricted 3rd Steiner root T. A contradiction follows as an S-restricted Steiner tree has no size 1 leaf representatives by definition. □

Therefore, from now on, we assume that every leaf node in $CC(G)$ has size at least 2.

Lemma 4.1.3. *Let G be a tree chordal graph with an S-restricted 3rd Steiner root T and u be an internal node in $CC(G)$. Suppose $r_1(u)$ and $r_2(u)$ are two distinct representatives for u. Then $d_T(r_1(u), r_2(u)) \leq 2$.*

Proof. Clearly, $d_T(r_1(u), r_2(u)) \leq 3$. Therefore assume that $d_T(r_1(u), r_2(u)) = 3$ and denote the path connecting r_1 and r_2 as r_1-x-y-r_2. Let v_1 and v_2 denote two nonadjacent nodes that are adjacent to u in $CC(G)$. For a representative $r(v_1)$ to be adjacent to both $r_1(u)$ and $r_2(u)$ it must be adjacent to x or y, as if it is adjacent to any other point it will be at a distance of at least 4. The same argument follows, by symmetry, for a representative $r(v_2)$. Therefore, $r(v_1)$ and $r(v_2)$ must be adjacent, a contradiction. □

Corollary 4.1.4. *Let G be a tree chordal graph with an S-restricted 3rd Steiner root T and let u be an internal node in $CC(G)$. Then, every vertex of the set of representatives for u is adjacent to a common point p in T.*

Lemma 4.1.5. *Let G be a tree chordal graph with an S-restricted 3rd Steiner root T and u be an internal node in $CC(G)$. Suppose $r_1(u)$ and $r_2(u)$ are two distinct representatives for u. If $d_T(r_1(u), r_2(u)) = 2$ and p is the node on the path between $r_1(u)$ and $r_2(u)$ in T, then only representatives of u are adjacent to p.*

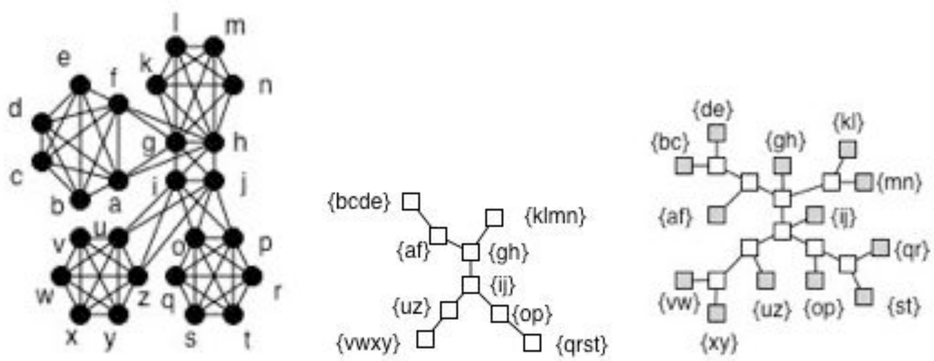
Proof. It follows that any representative r' adjacent to p will be adjacent to everything both $r_1(u)$ and $r_2(u)$ are adjacent to. Therefore, no other representative r^* can exist that is adjacent to $r_1(u)$, $r_2(u)$ and not r', implying that u is not internal, a contradiction. Therefore, only representatives of u or Steiner points are adjacent to p. □

Theorem 4.1.6. *Let G be a tree chordal graph such that every internal node in $CC(G)$ has size at least 2 and every leaf node has size at least 4. Then G has an S-restricted 3rd Steiner root tree.*

Proof. Starting with $CC(G)$, replace each node, representing a critical clique c_i, by a Steiner point s_i. If c_i is internal in $CC(G)$, create a single representative $r(c_i)$ and set it adjacent to s_i. If c_i is external in $CC(G)$, create two representatives $r_1(c_i)$ and $r_1(c_i)$, where $r_1(c_i)$ represents $\lfloor |c_i|/2 \rfloor$ vertices and $r_2(c_i)$ represents $\lceil |c_i|/2 \rceil$ vertices. Denote this new tree T.

As $CC(G)$ is a tree T is also a tree. We claim T is an S-restricted 3rd Steiner root tree. Every Steiner node has degree 3 and all representatives in T have size at least 2 by construction. The set S is empty, so the property holds vacuously. As adjacent nodes were at a distance of 1 in $CC(G)$, it follows that all representatives of adjacent critical cliques will be at distance of exactly 3 in T. Analogously, as nonadjacent nodes were at a distance of at least 2, nonadjacent nodes will be at a distance of at least 4 in T. Therefore T is an S-restricted 3rd Steiner root tree. □

An illustration of the construction process is in Figure 4.1, where Figure 4.1(a) shows G, Figure 4.1(b) shows $CC(G)$, Figure 4.1(c) shows an S-restricted 3rd Steiner root tree T for G.



(a) Graph G.

(b) $CC(G)$: labeled by the critical cliques.

(c) A 3rd Steiner root for $CC(G)$: representatives labeled by vertices they represent.

Figure 4.1: An example graph G shows the steps of operations for constructing a S-restricted 3rd Steiner root tree in the ideal case.

4.1.2 Structural Restriction 2

Theorem 4.1.6 deals with an ideal case where critical cliques must have sufficient size, namely internal critical cliques have size at least 2 and external critical cliques have size at least 4. The following several lemmas discuss the construction when there exist leaf nodes in $CC(G)$ of size less than 4.

Lemma 4.1.7. *Let G be a tree chordal graph with an S-restricted 3rd Steiner root T. Let $r(v)$ and $r(u)$ be representatives in T of two critical cliques in $CC(G)$. If $r(v)$ and $r(u)$ are adjacent a common Steiner point p then only representatives of u or v are adjacent to p.*

Proof. Assume, for contradiction, that a representative for another critical clique w is adjacent to x. Then u, v, and w form a C_3 in $CC(G)$, a contradiction to the assumption that $CC(G)$ is a tree. If adjacent to x is a Steiner point y which is adjacent to a representative for w, then again we have a C_3, a contradiction. Any representative that is further out will disconnect the graph. Therefore, no Steiner points or representatives are adjacent to x. □

Lemma 4.1.8. *Let G be a tree chordal graph with an S-restricted 3rd Steiner root T and let u be an internal node in $CC(G)$ with exactly two representatives $r_1(u)$ and $r_2(u)$. If $r_1(u)$ and $r_2(u)$ are adjacent then every internal neighbour v of u in $CC(G)$ has at most two representatives in any Steiner root T of G. Moreover,*

- *if v has exactly two representatives, then these two representatives must be adjacent in T, with one of them adjacent to either $r_1(u)$ or $r_2(u)$, and,*
- *if v has exactly one representative, then this representative must be adjacent to either $r_1(u)$ or $r_2(u)$.*

Proof. Suppose v has more than two representatives in T. By Lemma 4.1.4, all the representatives are adjacent to a common center point p. For both $r_1(u)$ and $r_2(u)$ to be adjacent to all representatives of v, one of them is adjacent to p, but this is a contradiction to v being internal, as nothing else can be adjacent to v and not u. Therefore, v has at most two representatives.

Assume v has exactly two representatives $r_1(v)$, $r_2(v)$ in T. $r_1(v)$ and $r_2(v)$ must be adjacent as otherwise, as above, $r_1(u)$ or $r_2(u)$ is adjacent to a common center point p of $r_1(v)$ and $r_2(v)$. It follows that since $r_1(u)$ is adjacent to $r_2(u)$ and $r_1(v)$ is adjacent to $r_2(v)$, they must form a path P_4, for all four to be within a distance of 3.

Assume v has exactly one representative $r(v)$ in T. If $r(v)$ is nonadjacent in T to $r_1(u)$ or $r_2(u)$, then as it must be within a distance of 3 from both we know that we have again a P_4. Assume without loss of generality, we have the path $r(v)-x-r_1(u)-r_2(u)$. Clearly, something must attach to x, otherwise x will be a degree 2 Steiner point. By Lemma 4.1.7, nothing but representatives of u or v can be adjacent to x. Therefore, $r(v)$ must be adjacent to either $r_1(u)$ or $r_2(u)$. □

Lemma 4.1.9. *Let G be a tree chordal graph with an S-restricted 3rd Steiner root T and let u be an internal node in $CC(G)$ with exactly two representatives $r_1(u)$ and $r_2(u)$, such that $d_T(r_1, r_2) = 1$. Then, u has at most one internal neighbor in $CC(G)$.*

Proof. From Lemma 4.1.8, if there are two internal neighbors v_1 and v_2, then one representative for v_1 must be adjacent to either r_1 or r_2 and one representative for v_2 must be adjacent to either r_1 or r_2. It follows that these two involved representatives for v_1 and v_2 are at distance either 2 or 3 in T, which contradicts the fact that v_1 and v_2 are not adjacent in $CC(G)$. □

Lemma 4.1.10. *Let G be a tree chordal graph with an S-restricted 3rd Steiner root T and let u and v denote two adjacent nodes in $CC(G)$. If u and v each have exactly one representative in T, say $r(u)$ and $r(v)$ respectively, then either $d_T(r(u), r(v)) = 1$ or $d_T(r(u), r(v)) = 3$. Moreover, if $d_T(r(u), r(v)) = 1$ then one of $\{u, v\}$ has degree at least 3 in $CC(G)$, and if $d_T(r(u), r(v)) = 3$, then both u and v are internal nodes in $CC(G)$.*

Proof. If $d_T(r(u), r(v)) = 2$ and x is the node adjacent to both of them, by Lemma 4.1.7 we have a contradiction. As $CC(G)$ contains at least 3 nodes, one of u or v must be internal, assume with out loss of generality u is internal. Let w be another node adjacent to u in $CC(G)$.

If $d_T(r(u), r(v)) = 1$ then the representatives for w are at distance exactly 3 from $r(u)$ in T, since they have to be at least 4 from $r(v)$. Let $r(w)$ be a representative for w, and the path connecting $r(u)$ and $r(w)$ be $r(u)$-x-y-$r(w)$. By the degree requirement, we need $deg_T(x) \geq 3$; therefore u must be adjacent to another critical clique that uses x as the path between its representatives. Therefore, node u has degree at least 3 in $CC(G)$.

If $d_T(r(u), r(v)) = 3$ and the path connecting $r(u)$ and $r(v)$ is $r(u)$-x-y-$r(v)$. As we require both $deg_T(x) \geq 3$ and $deg_T(y) \geq 3$, it follows that x and y must be attached to another point in T. As all leaves of T are representatives and as T is connected, it follows that $r(u)$ is within 3 from another representative; similarly for $r(v)$. Therefore, u and v are both internal. □

Lemma 4.1.11. *Let G be a tree chordal graph with an S-restricted 3rd Steiner root T and let u and v denote two adjacent nodes in $CC(G)$. If u has exactly one representative $r(u)$ in T and v has exactly two representatives $r_1(v)$ and $r_2(v)$ in T such that $r_1(v)$ and $r_2(v)$ are adjacent, then $r(u)$ is adjacent to either $r_1(v)$ or $r_2(v)$. Moreover, if v is a leaf node in $CC(G)$, then u has degree at least 3 in $CC(G)$.*

Proof. Note when both u and v are internal nodes in $CC(G)$ the result follows from Lemma 4.1.8. Assume with out loss of generality that $r(u)$ is closer to $r_2(v)$ than it is to $r_1(v)$. If $d_T(r(u), r_2(v)) = 2$, let $r(u)$-x-$r_2(v)$ be the path connecting $r(u)$ and $r_2(v)$. From Lemma 4.1.7, it follows that this is a contraction. Therefore, $r(u)$ is adjacent to $r_2(v)$.

If v is a leaf node in $CC(G)$, then u is internal in $CC(G)$. Let $r(w)$ be a representative for w, and the path connecting $r(u)$ and $r(w)$ be $r(u)$-x-y-$r(w)$. By the degree requirement, we need $deg_T(x) \geq 3$; therefore u be adjacent to another critical clique that uses x as the path between its representatives. Therefore, node u has degree at least 3 in $CC(G)$. □

Lemma 4.1.12. *Let G be a tree chordal graph with an S-restricted 3rd Steiner root T. Then, there is an S-restricted 3rd Steiner root for $CC(G)$ in which every leaf node in $CC(G)$ of size less than four has exactly one representative adjacent to one of the representatives for the neighboring node.*

Proof. Let u be a leaf node in $CC(G)$ of size less than 4. By Corollary 4.1.2, $2 \leq |u| \leq 3$. If $|u| = 2$, then u has exactly one representative, as it is a leaf in any 3rd Steiner root for $CC(G)$, by

Lemma 4.1.1. If $|u| = 3$, then at least one of the representatives $r_1(u)$ for u must appear as a leaf and $r(u) > 1$ in any 3rd Steiner root for $CC(G)$, by Lemma 4.1.1. If u has multiple representatives, the other representative $r_2(u)$ must have size 1. It follows as u is a critical clique and that $r_1(u)$ is a leaf in T, that we can delete $r_1(u)$ and increase the size of $r_2(u)$ to size 3.

Let $r(u)$ denote the unique representative for u and $r(v)$ be a representative for the neighbouring node v. As u is a leaf in $CC(G)$, v is the only adjacent representative. As by Lemma 4.1.7, we can not have a Steiner point between $r(u)$ and $r(v)$, it follows they are either adjacent or have two Steiner points on the path between them. In the latter case, it follows that as v is the only neighbour of u, the Steiner point adjacent to u will have degree 2. Therefore, they must be adjacent. □

Corollary 4.1.13. *Let G be a tree chordal graph with an S-restricted 3rd Steiner root T. Then there exist no degree 2 size 1 internal node u in $CC(G)$ such that u is adjacent to a leaf node v of size 2 or 3.*

Proof. By Lemma 4.1.12, it follows that there exists a Steiner root with a single representative $r(v)$ for v such that $r(v)$ is adjacent to the representative $r(u)$ for u. A contradiction follows as by Lemma 4.1.10, as the degree of u is 2. □

Lemma 4.1.14. *Let G be a tree chordal graph with an S-restricted 3rd Steiner root T. Then, there is an S-restricted 3rd Steiner root for $CC(G)$ in which every leaf node in $CC(G)$ of size at least four has exactly two representatives that are leaves in T which are separated by a non-representative Steiner node.*

Proof. Let u be such a critical clique. We can place the representatives of u, denote $r_1(u), r_2(u), ..., r_\ell(u)$, in one of three structural configurations:

1. $\ell = 1$;
2. $\ell > 1$ and all representatives adjacent to a common point p where p is either a Steiner point or representative of u; or,
3. $\ell > 1$ and all representatives adjacent to a common edge induced by points p_1 and p_2 where p_1 (p_2) is either a Steiner point or a representative of u.

If as in Case 1, we know by assumption that $|r_1(u)| \geq 4$ and, by Lemma 4.1.12, u must be adjacent to the representative $r(v)$ of another critical clique v. As u is a leaf v is its only neighbour. If $r(v)$ is the only representative for v, then replace $r_1(u)$ by two representatives $r'_1(u)$ of size $\lfloor |u|/2 \rfloor$ and $r''_1(u)$ of size $\lceil |u|/2 \rceil$. Create Steiner point p, and set it adjacent to $r'_1(u)$, $r''_1(u)$ and $r(v)$ adjacent to it. It trivially follows that this construction satisfies the lemma. If v has multiple representatives then replace $r(v)$ by a Steiner point p and append its vertices to another representative of v. Again, replace $r_1(u)$ with $r'_1(u)$ and $r''_1(u)$ and set them adjacent to p. Again, this is the postcondition from the lemma.

If as in Case 2, we are done if the common point is a Steiner point, else replace this common point p with a Steiner point and let the vertices p represents be represented by another representative of v. This satisfies the lemma.

If as in Case 3, either p_1 or p_2 is adjacent to a representative of u's only neighbour. Assume, without loss of generality that it is p_1. Then remove p_2 and the representatives adjacent to it other than p_1. Represent these representatives by p_1. We now continue as in Case 2. Thus, the lemma follows. $\square$

Lemma 4.1.15. *Let G be a tree chordal graph with an S-restricted 3rd Steiner root T. If u is an internal node in $CC(G)$ adjacent to $k \geq 2$ leaf nodes of size 2 or 3, then $|u| \geq k$ and there is a 3rd Steiner root T for $CC(G)$ such that there are at least k representatives for u that are adjacent to a common point. Moreover, the representative for each neighbouring leaf node of size 2 or 3 is adjacent to a distinct one of the representatives for u.*

Proof. From Lemma 4.1.12, we conclude that there is a 3rd Steiner root for $CC(G)$ in which there is exactly one representative for a leaf node of size 2 or 3 in $CC(G)$, and it is adjacent to a representative for the neighboring node, which is u in our case. It follows that there are at least k distinct representatives for u. Also, there are k representative that are adjacent to representatives of size-2 and 3 leaf nodes, and these k representatives are at distance exactly 2 to each other, that is, they are all adjacent to a common (Steiner or non-Steiner) point. $\square$

Corollary 4.1.16. *Let G be a tree chordal graph with an S-restricted 3rd Steiner root T. Then for every internal node u in $CC(G)$ the number of adjacent leaf nodes of size 2 or 3 is at most $|u|$.*

Corollary 4.1.17. *Let G be a tree chordal graph with an S-restricted 3rd Steiner root T. Then is no degree-2 size-2 internal node u in $CC(G)$, such that u is adjacent to a leaf node of size 2 or 3.*

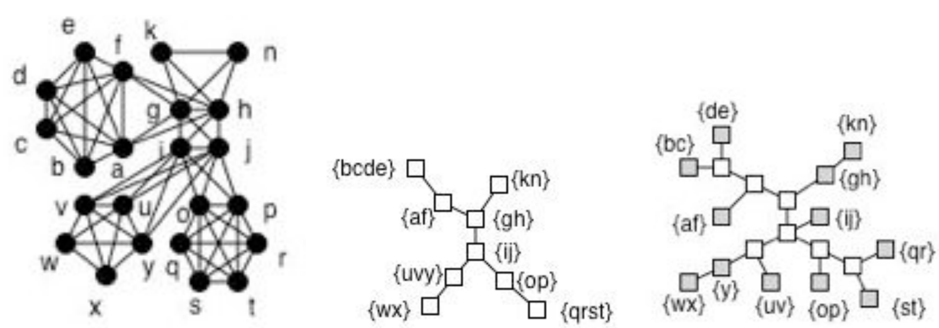
Proof. By Corollary 4.1.13, such a u can not have a single representative. Therefore, as it is size two, it must have two representatives $r_1(u), r_2(u)$ of size 1 that, therefore, must be internal in T. If follows from Lemma 4.1.7 that the leaf node representative is adjacent to $r_1(u)$ or $r_2(u)$ and they form an induced P_3. A contradiction follows from Lemma 4.1.10, which implies that u has degree at least 3. $\square$

Theorem 4.1.18. *Let G be a tree chordal graph such that in $CC(G)$ every internal node in $CC(G)$ has size at least 2 and every leaf node has size at least 2. Then if no degree 2 size 1 internal node is adjacent to a leaf node of size 2 or 3, every internal node u in $CC(G)$ has at most $|u|$ adjacent leaf nodes of size 2 or 3, and no degree-2 size-2 internal node in $CC(G)$ is adjacent to a leaf node of size 2 or 3 then G has an S-restricted 3rd Steiner root tree.*

Proof. From $CC(G)$, replace each node n_i in $CC(G)$ corresponding to critical clique c_i in G with a Steiner point p_i. For every internal node n_i adjacent to leaf node n_j such that $2 \leq |n_j| \leq 3$, place a representative $r(n_i)$ for a single vertex for n_i adjacent to p_i and p_j removing the existing edge. After considering all leaf representatives of n_i, represent all remaining vertices by a single representative adjacent to p_i. For all remaining internal nodes, represent all vertices by a single representative adjacent to p_i. For every leaf node n_i in $CC(G)$ of size at least 4, create two representatives of sizes $\lfloor |n_i|/2 \rfloor$ and $\lceil |n_i|/2 \rceil$, respectively, and attach them to p_i. Finally, for every leaf node n_i in $CC(G)$ of size 2 or 3, replace its the Steiner point p_i with a single representative. This constructs an S-restricted 3rd Steiner root T for $CC(G)$

In T, every leaf node of size 2 or 3 has exactly one representative adjacent to a representative of its only neighbour, satisfying Lemma 4.1.12. Leaf nodes of size four or more have two representatives adjacent to a Steiner point of degree at least 3. Internal nodes which are not adjacent to any leaf nodes of size 2 or 3, are represented by a single vertex adjacent to a Steiner point of degree at least 3. Internal nodes, adjacent to a leaf node of size 2 or 3, have all representatives adjacent to a common Steiner point. As G satisfies Corollaries 4.1.2, 4.1.13 and 4.1.16 it follows that the degree of this Steiner point is at least 3. As nodes in $CC(G)$ were adjacent if and only if at distance of 1, it follows that adjacent nodes are either at distance of at most 3 in T and nonadjacent node are at distance of at least 4. $\square$

An illustration of the construction process is in Figure 4.2, where Figure 4.2(a) shows G, Figure 4.2(b) shows $CC(G)$, and Figure 4.2(c) shows an S-restricted 3rd Steiner root for $CC(G)$.



(a) Graph G.

(b) $CC(G)$: labeled by critical cliques.

(c) An S-restricted 3rd Steiner root for $CC(G)$: representatives labeled by vertices they represent.

Figure 4.2: An example graph G shows the steps of operations for constructing a S-restricted 3rd Steiner root tree T when leaf nodes of size 2 and 3 exist.

Theorem 4.1.19. *Let G be a tree chordal graph such that every internal node in $CC(G)$ has size at least 2 and every leaf node has size at least 2. If G satisfies Corollaries 4.1.13, 4.1.16, and 4.1.17, then an S-restricted 3rd Steiner root tree can be constructed in linear time.*

Proof. Using Theorem 4.1.18, checking if G is tree chordal and, if so, building the critical clique graph $CC(G)$ takes linear time by Corollary 2.2.9. Corollaries 4.1.13, 4.1.16, and 4.1.17 can be checked in linear time by examining each node and checking that it satisfies each of the conditions. The replacement of the nodes in $CC(G)$ with Steiner points and setting the vertices adjacent to these Steiner points we show is also linear. As the number of nodes in $CC(G)$ is bounded above by the number of vertices in G this replacement takes at most $O(|V|)$ time. For each critical clique we then, using the corresponding lemma, place its vertices adjacent to the corresponding Steiner point; as we are constructing a tree this is an addition of $|V|$ edges and $|V|$ vertices. Therefore, we do a constant amount of work to each of the $O(|V|)$ vertices, implying an overall linear runtime. □

4.1.3 k-PRP Algorithm for Tree Chordal Graphs

We now present the remaining details of the algorithm, namely we deal with the existence of size 1 internal nodes in $CC(G)$. Clearly, there is a unique representative for each such internal node u and the representative $r(u)$ must be internal in any 3rd Steiner root for $CC(G)$.

Lemma 4.1.20. *Let G be a tree chordal graph with an S-restricted 3rd Steiner root T and let u denote a size-1 internal node in $CC(G)$. Then either $CC(G)$ has exactly 3 nodes or u has degree at least 3 in $CC(G)$. Moreover, if u has degree exactly 3, then one of its neighboring nodes must have size greater than the number of adjacent leaf nodes of size 2 oor 3 in $CC(G)$.*

Proof. By Lemma 4.1.10, we note that u is adjacent to at most one leaf of size 2 or 3, and if it is, then u must necessarily have degree 3 in $CC(G)$. Therefore, assume u has degree 2 and is not adjacent to any leaf node of size 2 or 3. Let $r(u)$ denote the unique representative for node u. Let v and w be the two adjacent nodes of u. The representatives of v and w must be at a distance of at least 4 from each other, therefore, at most one has a representative adjacent to $r(u)$. Without loss of generality assume $r(v)$ is adjacent to $r(u)$, then all the representatives of w are at a distance of three from $r(u)$. Let $r(u)$-x-y-$r(w)$ denote one of these such paths. We note, by Lemma 4.1.7, only Steiner points of u and w can be adjacent to x and y. But the degree of x must be at least 2, and as u has only a single representative, it must be a representative of w, thus, a contradiction to the distance being exactly 3.

Therefore, the representatives of v and w must be at a distance of exactly 2 from $r(u)$. Denote one such path $r(v)$-x-$r(u)$-y-$r(w)$. By Lemma 4.1.7 and as u has only single representative, only representatives of $r(v)$ (respectively $r(w)$) can be adjacent to x (respectively y). Therefore, both v and w must have at least two representatives attached to x and y, respectively. Therefore G must consist of exactly three nodes, as nothing can be adjacent to w and v without being adjacent to u, contradicting u having degree 2. Therefore, u must have degree at least 3 in $CC(G)$.

When u has degree exactly 3 in $CC(G)$, $CC(G)$ could have 4 nodes where all nodes other then u have size at least 4, as the above case with 3 nodes. If all nodes are at distance three than as u must be internal we will have the situation of a path $r(u)$-x-y-$r(w)$, implying that $r(u)$ has degree at least 4. Therefore, there must be a representative $r(v)$ adjacent to $r(u)$ in the root. It follows that no representative for other nodes than u and v could be adjacent to $r(v)$. For each leaf node adjacent to v of size 2 or 3, by Lemma 4.1.12, we must have a representative of v adjacent to the representative of the leaf. Therefore, as one representative is adjacent to $r(u)$, it follows that $|u|$ is strictly greater than the number of adjacent nodes of size 2 or 3 is $CC(G)$. □

Lemma 4.1.21. *Let G be a tree chordal graph and let u be a size 1 internal node in $CC(G)$ with exactly one leaf node v of size 2 or 3 adjacent to u in $CC(G)$. Let G' be the graph created by removing v and increasing the size of u to 2. Then G' has an S-restricted 3rd Steiner root tree T' if and only if G has an S-restricted 3rd Steiner root tree T.*

Proof. Let G have an S-restricted 3rd Steiner root tree T. Then there is a unique representative $r(u)$ for u and a unique representative $r(v)$ for v. Moreover, $r(v)$ is a leaf in the associated 3rd Steiner root tree for $CC(G)$ and no representatives for nodes other than u can be within distance 3 to $r(v)$. Therefore, if we remove $r(v)$ and increase the size of $r(u)$ to two, the new tree T' is a valid S-restricted 3rd Steiner root tree corresponding to the graph G' as modified in the lemma.

Let G' have an S-restricted 3rd Steiner root tree T'. By the construction of Theorem 4.1.18, Represent u by a single vertex adjacent to a Steiner point p. As no leaf node of size 2 or 3 is adjacent to u, it also follows that only Steiner points are adjacent to p. Therefore, all neighbouring representatives are at distance of exactly 3. Therefore, let $r(u)$ only represent a single vertex and place a representative, corresponding to v, or size 2 or 3 adjacent to $r(u)$. It follows that this new tree T is a valid S-restricted 3rd Steiner root tree corresponding to the graph G as in the lemma. □

Lemma 4.1.22. *Let G be a tree chordal graph and let u be a size 1 internal node in $CC(G)$ with exactly one leaf node v adjacent to u in $CC(G)$ and $|v| \geq 4$. Let G' be the graph created by removing v and increasing the size of u to 2. Then G' has an S-restricted 3rd Steiner root tree T' if and only if G has an S-restricted 3rd Steiner root tree T.*

Proof. Let G have an S-restricted 3rd Steiner root tree T; it follows from Lemma 4.1.14 and its proof that there is a phylogenetic root for G such that there are exactly two representatives for node v that are separated by a Steiner point p and p is adjacent to the unique representative $r(u)$ for node u. Consequently, we can remove the two representatives for v, p, and increase the size of $r(u)$. It follows that this new tree T' is a valid S-restricted 3rd Steiner root tree corresponding to the graph G as in the lemma.

Let G' have an S-restricted 3rd Steiner root tree T'. By the construction of Theorem 4.1.18, represent u by a single vertex adjacent to a Steiner point p; therefore, all neighbouring representatives are at distance of at least 2. Therefore, let $r(u)$ only represent a single vertex and place a Steiner point p adjacent to $r(u)$. Create two representatives for v of suitable size and set them adjacent to p. It follows that this new tree T is a valid S-restricted 3rd Steiner root tree corresponding to the graph G as in the lemma. □

Lemma 4.1.23. *Let G be a tree chordal graph and let u be a size 1 internal node in $CC(G)$ with exactly no leaf nodes adjacent to u in $CC(G)$. Let G' be the graph created by increasing the size of u to 2. Then the graph G' has an S-restricted 3rd Steiner root tree T' if and only if G has an S-restricted 3rd Steiner root tree T.*

Proof. Let G have an S-restricted 3rd Steiner root tree T. Trivially, $T' = T$ is a valid S-restricted 3rd Steiner root tree for G'.

Let G' have an S-restricted 3rd Steiner root tree T'. From Lemma 4.1.20, u has at least three neighbors in $CC(G)$. By the construction of Theorem 4.1.18, u has exactly one representative $r(u)$ that appears as a leaf in the associated 3rd Steiner root for $CC(G')$. Moreover, $r(u)$ is at distance exactly 3 to any other representatives for the neighboring nodes to u. If u has all non-neighbours at distance of 5 or more in T, then replace the Steiner point p adjacent to $r(u)$ with $r(u)$ and reduce its size to 1. Assume then, that there exists a non-neighbour at distance 4 from $r(u)$. By the construction of the theorem, it follows that adjacent to p is the Steiner points adjacent to the representatives of all of the neighbours of u. Therefore, if u has 4 or more neighbours then by splitting p into two Steiner nodes p_1 and p_2 such that each of them inherits at least two edges, and removing one vertex from u, we obtain a S-restricted 3rd Steiner root tree T for G. If u has degree of 3 in $CC(G)$, from Lemma 4.1.20 we have at least one neighbouring node v to u such that its size is larger than the number of adjacent leaf nodes of size 2 and 3 in $CC(G)$. Consequently, either we have one representative for v that is a leaf in the Steiner root, or we can create a new representative $r(v)$ for v and make it adjacent to the Steiner point, $p(v)$ of v. The obtained tree is no longer an S-restricted 3rd Steiner root tree T, but by removing edge between $p(u)$ and $p(v)$ and adding edge between $r(u)$ and $r(v)$ it becomes a valid S-restricted 3rd Steiner root tree T. Furthermore, we may reduce the size of u from 2 to 1 and the resultant tree is a S-restricted 3rd Steiner root tree T. □

Theorem 4.1.24. *Let G be a tree chordal graph G. Then there exists a linear time algorithm to decide whether G has an S-restricted 3rd Steiner root tree T, and if so, return such a T.*

Proof. First of all, if the critical clique graph $CC(G)$ contains only one or three nodes, then one can determine G has an S-restricted 3rd Steiner root tree T is trivial. Assume $CC(G)$ contains more than three nodes. Determining if G does not have an S-restricted 3rd Steiner root tree T can be done by check the following conditions:

1. $CC(G)$ contains no size 1 leaf nodes (Corollary 4.1.2);

2. Every size-1 internal node in $CC(G)$ has degree at least 3; and if it has degree exactly 3 then one of its neighboring node must have size greater than the number of its adjacent leaf nodes of size 2 and 3 in $CC(G)$ (Corollary 4.1.13 and Lemma 4.1.20).

3. Every internal node in $CC(G)$ has size at least as large as the number of adjacent leaf nodes of size 2 and 3 (Corollary 4.1.16);

4. There is no degree 2, size 2, internal node c in $CC(G)$ that is adjacent to a leaf node of size 2 or 3 (Corollary 4.1.17);

Create the modified graph G' by following Lemmas 4.1.21, 4.1.22 and 4.1.23, this increases every size-1 internal node to have size 2. We then can apply Theorem 4.1.18 to construct an S-restricted 3rd Steiner root tree T' for the modified graph G', and finally according to Lemmas 4.1.21, 4.1.22 and 4.1.23 to construct an S-restricted 3rd Steiner root tree T for the given graph G. Note that conditions 2–4 guarantee graph G to have an S-restricted 3rd Steiner root tree T if and only if the modified graph G' has an S-restricted 3rd Steiner root tree T'.

Checking if G is tree chordal and, if so, building the critical clique graph $CC(G)$ takes linear time by Corollary 2.2.9. The four conditions above can be checked in linear time by examining each node and checking that they satisfy each of the conditions. Producing the modified graph G' requires finding all size 1 nodes in $CC(G)$, possibly delete a neighbour, and increasing the size to 2; each of these check is constant time, therefore, creation of G' takes linear time. By Theorem 4.1.19, we can produce a S-restricted 3-root Steiner tree T' for the graph G' in linear time. Finally, we find the tree T by modifying T'. As we add a constant bounded amount to each size 1 node in G, this again is linear. The overall construction is linear as $V(T) \leq 2V(G)$ and $E(T) \leq 2E(G)$. This proves the theorem. □

By Theorem 4.1.24 and Lemma 3.1.1 we have the following corollary.

Corollary 4.1.25. *Let G be a tree chordal graph G. Then there exists a linear time algorithm to decide whether G has a 5th phylogenetic root tree T, and if so, return such a T.*

4.2 Decomposition Construction

Let G be a strictly chordal graph with an S-restricted 3rd Steiner root tree T, where S is the set of all size 1 critical cliques in G. Let $\mathcal{T}$ be a forest of tree chordal graphs decomposed from G. Let c be a critical clique contained in a large maximal clique in G and contained in a tree chordal graph T_i, decomposed in to at least two nodes in $CC(T_i)$.

We now consider the construction of an S-restricted 3rd Steiner root tree T_i' for a T_i in $\mathcal{T}$ as an intermediary step in the process of the 5-PRP algorithm for strictly chordal graphs, therefore, we will allow a critical clique such as c to be adjacent to a degree 2 Steiner point or c to have a single size 1 leaf representative. These inconsistencies are allowable as long as they do not exist in the final S-restricted 3rd Steiner root tree T for G.

We first observe that if c has a single size 1 leaf representative r_c in T_i' then r_c must be internal in T, by Corollary 4.1.2. As we will see in Chapter 5, this is always able to be done.

For the degree 2 Steiner point, we note that c must have a single representative r_c as it is adjacent to another critical clique in T_i. r_c will need to be adjacent to an additional Steiner point in T, as the degree must be at least 3 in T and if another representative is adjacent to this Steiner point then it will be indistinguishable from r_c.

Therefore, for the follow S-restricted 3rd Steiner root tree constructions we allow critical cliques such as c to be adjacent to a degree 2 Steiner point or to have a single size 1 leaf representative. As such we define the set $\mathcal{C}$ to be the critical cliques that are contained in large maximal cliques of G. Let S be the set of critical cliques of size 1 in G. For a $T_i \in \mathcal{T}$, we define the sets

$$W_i = \{c_i \mid c_i \in S \setminus \mathcal{C} \;\&\; c_i \in V(T_i)\},$$

$$X_i = \{c_i \mid c_i \in \mathcal{C} \;\&\; c_i \in V(T_i)\}.$$

We will now show constructions to produce an S-restricted 3rd Steiner root tree T for a tree $T_i \in \mathcal{T}$, such that $S = W_i$ and a Steiner point adjacent to critical clique $c \in X_i$ can have degree 2.

4.2.1 Trivial Tree Chordal Graphs

We first deal with when $CC(G)$ contains less than 3 nodes - trivial tree chordal graphs. A tree chordal graph T is *trivial* when CC(T) is a single node. No connected graph will have a $CC(G)$ with two nodes; as the two adjacent critical cliques would be one large critical clique.

Trivial tree chordal graphs T can arise in two ways, when decomposed from a strictly chordal graph: T was part of only large maximal cliques in G, or T was part of a large maximal clique and decomposed into a tree chordal graph of exactly two critical cliques. In the second case, we distinguish the two critical cliques as they have a different neighbourhood in G. Therefore, we describe an algorithm to handle trivial tree chordal graphs.

Corollary 4.1.2 can easily be adapted to show that no size 1 leaf exists in trivial tree chordal graphs. Lemma 4.1.3 shows that a critical clique has diameter at most 2 in the Steiner tree. Finally, Lemma 4.1.12 shows that leaf nodes of size 2 or 3 are are represented by a single vertex adjacent to a neighbours representative and Lemma 4.1.14 shows that leaf nodes of size at least 4 can all be represented by two representatives adjacent to a common Steiner point. Using these results, Figure 4.3 shows all possible S-restricted 3rd Steiner root trees for three three types of trivial chordal graphs.

A critical clique c is *constrained* in K if c has two or more representatives in the Steiner tree T, c has a single representative adjacent to a Steiner point with Steiner degree 1, or c has a single representative adjacent to the representative of another critical clique. As shown in Section 5.1.1, at most one critical clique can be constrained in a large maximal clique of a strictly chordal graph with an S-restricted 3rd Steiner root tree. Therefore, the algorithm aims to produce constrained critical cliques only when necessary.

If a tree chordal graph was part of only large maximal cliques in G, the corresponding S-restricted 3rd Steiner root tree to the T_i will be a single representative. See Figure 4.3(a) for an enumeration of all possible trees; notice that the single vertex is common in all possibilities and is unconstrained for each of these. Note that $S = \emptyset$, as W_i does not contain this critical clique.

For the second case, we present the following algorithm $TrivAlg(T_i)$.

1. if both c_1 and c_2 are internal critical cliques in G then represent c_1 and c_2 by two single representatives connected by a path of two Steiner points; or,

2. if only one is an internal critical clique, assume c_1, then:

   a. if $|c_1| = 1$ and $1 < |c_2| < 4$ then represent c_1 and c_2 by two single adjacent representatives;

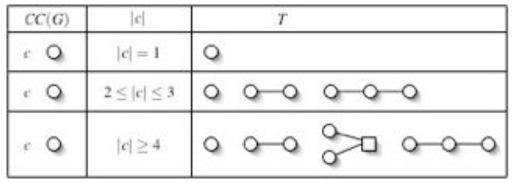
   b. if $|c_1| > 1$ and $1 < |c_2| < 4$ then represent c_1 by r'_{c_1} and r''_{c_1}, c_2 by r_{c_2}, and create path $r'_{c_1} - r''_{c_1} - r_{c_2}$;

   c. if $|c_2| > 3$ then represent c_1 by a single representative, represent c_2 by two representatives of sizes $\lceil |c_2|/2 \rceil$ and $\lfloor |c_2|/2 \rfloor$, and make all adjacent to a common Steiner point; or,

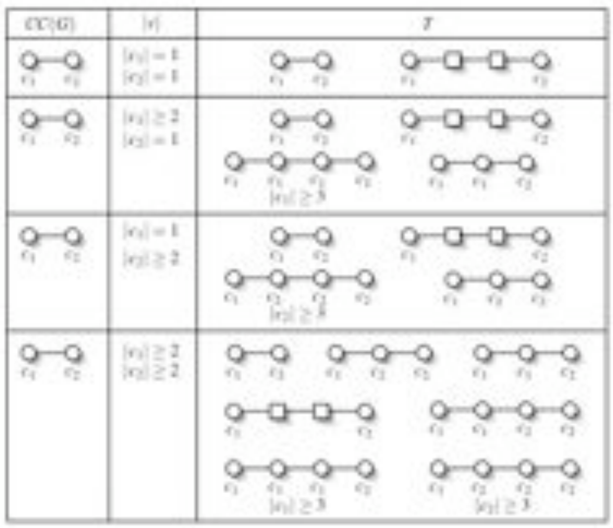
   d. otherwise ($|c_2| = 1$) no S-restricted 3rd Steiner root tree exists.

The trees produces by $TrivAlg(T_i)$ satisfy the condition of being an S-restricted 3rd Steiner root tree for T_i with respect to W_i and X_i. Figure 4.3(b) corresponds to the possible choice for Case 1. Notice that all configurations for this critical clique are constrained. The choice for critical cliques represented by a path of representatives are not chosen in this case as all adjacent critical cliques will need to have single representatives and, as we will show, one possibility will force one maximal cliques contained critical cliques to have single representatives. Therefore, we choose the less restrictive case. This leaves two options for the S-restricted 3rd Steiner root tree: (1) the option presented in the algorithm and (2) letting c_1 and c_2 be adjacent, with no Steiner points. For 1) c_1 and c_2 must be contained in two additional maximal cliques each in G, one adjacent to c_1 or c_2 (if $|c_1| = 1$ or $|c_2| = 1$) and the other adjacent to the Steiner points adjacent to c_1 and c_2. For (2) that c_1 and c_2 must also be contained in two additional maximal cliques each. The difference is all maximal cliques adjacent to c_1 and c_2 will have to all adjacent critical cliques in $CC(G)$ as unconstrained for (2), whereas only one maximal clique needs all contained critical cliques as unconstrained for (1) (See Lemma 5.2.1).

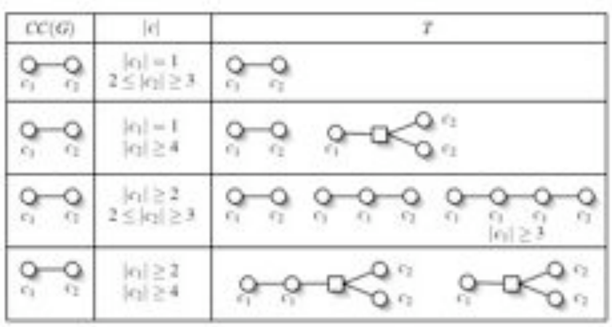
For similar reasons, the optimal cases are chosen from the cases of Figure 4.3(b). Cases 2a and 2b satisfy Lemma 4.1.12, where case 2a must be contained in at least two additional maximal cliques in G and case 2b is only contained in at least one. Case 2c is ideal with no restriction place on the maximal cliques adjacent to c_1. For Case 2d no S-restricted 3rd Steiner root tree exist for G, as c_2's representative will always be external and have size 1.



(a) Critical clique graph has exactly one node.



(b) Critical clique graph has exactly two nodes. Both c_1 and c_2 are internal in $CC(G)$.



(c) Critical clique graph has exactly two nodes. c_1 is internal in and c_2 is external in $CC(G)$.

Figure 4.3: All possible S-restricted Steiner trees corresponding to the three types of trivial tree chordal graphs.

4.2.2 Tree chordal graphs

We now describe and justify a modification of the tree chordal algorithm of Section 4.1.3 to minimize constrained critical cliques.

Given a tree chordal graph $T_i \in \mathcal{T}$ decomposed from a graph G, set S corresponding to nodes in T_i of size 1 in $CC(G)$, and a set R corresponding to nodes of $CC(T_i)$ contained in maximal cliques of size three or more in $CC(G)$ produce an S-restricted 3rd Steiner root tree as follows. Denote this modified algorithm $ALG(G)$, where G is a tree chordal graph.

- If T_i was part of only large maximal cliques in G, return a single representative.
- If T_i was part of a large maximal clique and decomposed into a tree chordal graph of exactly two critical cliques, return tree as in $TrivAlg(T_i)$.
- Produce tree chordal graph $T_i^\star$ as follows:
  - size two and three external nodes contained in R, change size to four;
  - size one external nodes contained in R adjacent to degree-2 size-2 node in $CC(T_i)$, change size to four;
  - remaining size one external nodes contained in R, change size to two; and,
  - size one internal nodes contained in R which are not adjacent to an external node not contained in R, change size to two.
- Call the tree chordal algorithm with the modified tree $T_i^\star$.
- return no if the tree chordal algorithm fails, or return the S-restricted 3rd Steiner root tree.

Lemma 4.2.1. *Given a strictly chordal graph G decomposed into a forest of tree chordal graphs $\mathcal{T}$ and set S corresponding to nodes in G of size 1, if $ALG(T)$ fails to produce a valid S-restricted* 3rd *Steiner root tree for any $T_i \in \mathcal{T}$ then no S-restricted* 3rd *Steiner root tree exists for G.*

Proof. $TrivAlg(T)$ only rejects when $|c_2| = 1$, as $c_2 \in S$ and will never be internal.

To show the correctness of the modified tree chordal algorithm we prove the contrapositive; assume G has an S-restricted 3rd Steiner root tree T', we will show for any tree chordal graph $T_i \in \mathcal{T}$ produced in the decomposition, the modified algorithm will produce an S-restricted 3rd Steiner root tree. Take any $T_i \in \mathcal{T}$ in such a graph G. Denote T_i' as the subtree induced on the tree T' with the representatives of the critical cliques of T_i and the Steiner points on paths between these representatives. We show from T_i' a valid S-restricted 3rd Steiner root tree for the T that $ALG(T_i)$ produces.

First, as T' satisfies all distance requirements, it follows that so will T_i'. All representatives that are not part of large maximal cliques in G will satisfy the size requirements; all Steiner points not adjacent to a representative of a critical clique in large maximal clique will satisfy the minimum degree requirement. Therefore, assume that c is a critical clique in T_i' that fails; it follows either a representative of c is external in T_i' and has size one or a Steiner point adjacent to c has degree two.

We first deal with the case when an external representative is size one. The modifications to $ALG(G)$ set external representatives of size one to size two. Therefore, c must have two representatives. But, the other representative was a leaf in T' implying its size is greater than one. This implies the total size of this critical clique was at least three, but external critical cliques of size three were modified to have size four. Therefore, we can represent these nodes with two representatives of size two each in T_i'.

If the Steiner point adjacent to c has degree two, it must of been adjacent to a Steiner point of the maximal clique. The representative of the critical clique c must also have size one, as size greater than two were modified to have size four or more and could have two representatives. This could be the case when the critical clique c is internal as well. By as in the previous case, this implies another representative must exist and we modified critical cliques of size 2 or 3 to size four and, therefore, this critical clique can have two representatives.

□

By Lemmas 3.1.1 and 4.2.1, if $ALG(T)$ fails for any tree chordal graph, we can return no, as no S-restricted 3rd Steiner root tree exits and therefore no 5th phylogeny root tree will exist. We now enumerate the possibilities of a critical clique returned by $ALG(T)$.

Lemma 4.2.2. *Given a tree chordal graph G with at least two critical cliques, $ALG(G)$ leaves the representatives of any critical cliques in the 3rd Steiner root tree T in exactly one of the follow states:*

- *c1: Representatives adjacent to a Steiner point of Steiner degree one; nearest representative of another critical clique is at distance of three with:*
  - *a: a single representative, or*
  - *b: two representatives,*
- *c2: One representative adjacent to a degree two Steiner point, with:*
  - *a: nearest representative of another critical clique is at distance of three, or*
  - *b: nearest representative of another critical clique is at distance of two,*
- *c3: One representative at distance of one to another leaf critical clique and a Steiner point, other critical cliques are at a distance of three,*
- *c4: One representative adjacent to one a representative of another critical clique.*
- *c5: Two adjacent representatives; one adjacent to another leaf's representative.*

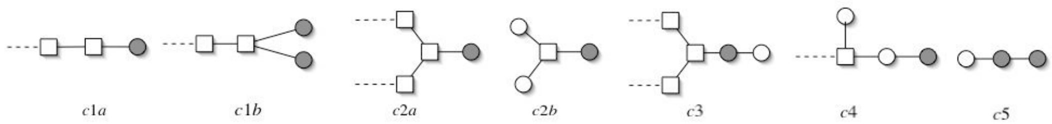


Figure 4.4: The seven possible cases for a leaf critical clique in a tree chordal graph as returned by algorithm *ALG*; representatives are darkened. We denote these cases as $c1a$, $c1b$, $c2a$, etc. Notice only $c2a$ and $c2b$ are unconstrained.

Proof. The following correspondences are for the trivial tree chordal graphs in $TrivAlg(T)$: both critical cliques in case 1 will correspond to $c1$, case 2a will correspond to $c4$, case 2b will correspond to $c5$, and the case 2c will correspond to $c2b$.

As the $ALG(G)$ modifies all size one internal nodes to size two, Theorem 4.1.18 gives internal nodes of size at least two a single representative ($c2a$), external nodes of size 1, 2 or 3 that were modified to have four representatives ($c1a$) and external nodes of size at least four by two representatives adjacent to a single Steiner point ($c1b$) in a 3-root Steiner tree. Increase the size of all size one external nodes to size two; Lemma 4.1.12 leaves a size two external node as a single representative adjacent to a representative of an internal critical clique ($c4$). The algorithm modifies all size two or more external critical cliques to size at least four; in Theorem 4.1.18 these will have two representatives adjacent to a Steiner point as in $c2a$. $\square$

Chapter 5

5-root Phylogeny Tree Construction for Strictly Chordal Graphs

As shown in the previous chapter, we have 7 possibilities for critical cliques returned by the tree chordal algorithm. This chapter[1] shows how to deal with the possible configurations that could be in any large maximal clique.

5.1 Preliminaries

Before we present the algorithms, we first discuss structure of large maximal cliques (Section 5.1.1). We will present the algorithm in three progressively less restrictive parts. Section 5.2.1, will assume G contains no small leaves and at most one critical cliques is constrained. Section 5.2.2 will restrict G to not contain small leaves. Section 5.2.3 will show the entire construction for strictly chordal graphs.

5.1.1 Structure of large maximal cliques

The following lemma, Lemma 5.1.1, is an example of structure that is a potential problem for construction of an S-restricted 3rd Steiner root tree; the following section shows why this poses a problem and how it becomes unnecessary in the construction of an S-restricted 3rd Steiner root tree.

Lemma 5.1.1. *[24] Let G be a graph with a 3rd Steiner root T. Assume there exist in G three maximal cliques K_1, K_2, K_3 such that $K_1 \cap K_2 = I_1 \neq \emptyset$, $K_2 \cap K_3 = I_3 \neq \emptyset$, and $K_1 \cap K_3 = \emptyset$. Let $I_2 = K_2 - I_1 - I_3$. If $I_1 = \{u_1, u_1'\}$, $I_3 = \{u_3, u_3'\}$, and $|I_2| > 0$, then u_1-u_1'-u_3'-u_3 is a path in T and every representative for a critical clique in I_2 is adjacent to either u_1' or u_3'.*

The above lemma has all critical cliques as constrained as all the critical cliques contained in I_2 are as $c4$ and the critical cliques I_1 and I_3 are as in $c5$. As we will show in Lemma 5.1.3, we can maintain all adjacencies while changing this maximal clique K_2 to have all critical cliques as unconstrained.

[1] A version of this chapter has been submitted for publication. Kennedy and Lin 2005. ISAAC [23]

Lemma 5.1.2. *[24] Let G be a graph with a 3rd Steiner root T, then each maximal cliques with critical clique cardinality of 3 or more either has exactly two critical cliques each with two representatives as in Lemma 5.1.1, or has at most one internal critical clique with two are more representatives in T.*

Proof. Let G be a graph and let T be its 3rd Steiner root tree. All representatives of critical cliques contained in a maximal clique of G are either adjacent to a single point in T or are adjacent to one of two adjacent points v_1, v_2 in T.

In the first case, an external critical clique can be adjacent to at most two representatives in T, where if exactly two, one must be the single central point. Therefore, all critical cliques except at most one have a single representative.

In the latter case, if v_1, v_2 are part of the maximal clique then it is exactly the situation in lemma 5.1.1. If exactly one of v_1, v_2 is a Steiner point then we would have a path u_1-s-u_3'-u_3 where other critical cliques are adjacent to either s or u_3'. All except at most one critical clique is a leaf critical clique and a critical clique adjacent to s or u_3' can never be adjacent to a single critical clique. Therefore in order for a critical clique to be adjacent to a single critical clique it follows that all critical cliques are represented by a single representative other than u_3 and u_3. If both v_1, v_2 are Steiner points, since there are more than three critical cliques, at most one side can have multiple representatives for the same critical clique. $\square$

Lemma 5.1.3. *Let K be a maximal clique represented by the situation of Lemma 5.1.1 in a 3rd Steiner root T, then there exists an equivalent representation with a central Steiner point adjacent to the representatives of its critical cliques in K.*

Proof. Given the structure as in Lemma 5.1.1 identify u_1 and u_1' into one representative, u_1^*, analogously with u_3 and u_3', produce u_3^*. Create a new Steiner point s and make u_1^*, u_3^* and all representatives for I_2 adjacent to s. Notice that since all critical cliques that are adjacent to exactly I_1 are at a distance of 3 from i_1' are now that distance from s. The same follows for all critical cliques adjacent to i_3' and critical cliques in I_2. Therefore, the new structure is equivalent to the old tree. $\square$

The following corollary follows easily from Lemmas 5.1.2 and 5.1.3.

Corollary 5.1.4. *Let G be a graph with a 3rd Steiner root T. Then there exists a representation in which all maximal cliques with critical clique cardinality of 3 or more have at most one internal critical clique with two or more representatives.*

5.2 5PRP on Strictly Chordal Graphs

This section deals with the combination of the Steiner trees returned by $ALG(G)$ and progresses from the most trivial case to the complete case: the solution of the 5-root phylogeny problem on strictly chordal graphs.

5.2.1 Structural Restriction 1

A *small leaf* is an external critical clique of size 1 in a maximal clique of critical clique cardinality at least three. We remind the reader that a constrained critical clique in a maximal clique K is a critical clique such that either it has two or more representatives in the Steiner tree T, it has a single representative adjacent to a Steiner point with Steiner degree 1 in T, or it has a single representative adjacent to the representative of another critical clique in T. In the following section we will assume that the input graph G contains no small leaves and large maximal cliques K contain at most one critical clique which constrained. Therefore, at most one critical clique in a large maximal clique will be as $c1a$, $c1b$, $c3$, $c4$, or $c5$. $c4$ is a very restrictive case as the following lemma shows.

Lemma 5.2.1. *Let G be a graph with an S-restricted 3rd Steiner root tree T and a maximal clique K. If K has an internal critical clique as in $c4$ then the critical clique must be part of at least two maximal cliques with critical clique cardinality three or more with other critical cliques unconstrained.*

Proof. The unique representative r of the critical clique in $c4$ is adjacent to another critical clique's representative. Therefore, any large maximal clique K that contains the critical clique has the representatives of all critical cliques contained in K other than r at distance 3 from r in T. This implies, that each of these critical cliques is unconstrained and a path of two Steiner points must be between r and these critical cliques. The size of r must be 1, as if had size 2 or more it would have been as in $c1a$ by $ALG(G)$, thus r necessarily has a single representative. Therefore, there exists a representation where all paths between r and the adjacent critical cliques share the Steiner point directly adjacent to r as their path to r, as otherwise, the Steiner point adjacent to r will have degree 2. □

The structure of a critical clique in case $c1a$, is a single representative adjacent to a Steiner point with Steiner degree one; as such, if the corresponding critical clique c has size 1 then we must increase the degree of both this representative and the Steiner point. It follows that c must be part of at least three maximal cliques; the following operation shows how to increase the degree of both the Steiner point and the representative of c.

Definition 5.2.1 (Operation 1). *Let c be a critical clique part of at least three maximal cliques $K_1, K_2, ..., K_n$. If K_1 is in $c1a$ or $c2b$ and K_2 has all critical cliques unconstrained then assign c a single representative and let the Steiner point adjacent to c in K_1 be adjacent to the Steiner point from K_2 such that all its critical cliques are at distance exactly three from c. For $K_3, ..., K_n$, c now corresponds to $c2a$ and is unconstrained.*

If a critical clique needs to have Operation 1 performed, check that there exists a maximal clique containing it that has all critical cliques unconstrained. If no maximal clique exists, we check if an adjacent critical clique is as $c1a$ or $c1b$, and apply Operation 1. In a similar fashion continue searching for a resolvable path through maximal cliques. Note that such a search is a depth first search through the tree, and in the worst case, and has a linear runtime. Notice that the choice made to change a path by Operation 1 will never affect another path as the search will assign a single representative for a critical clique, and this critical clique will be now unconstrained. Therefore, we pick the first resolvable path.

Theorem 5.2.2. *Let G be a connected strictly chordal graph G such that G contains no small leaves and large maximal cliques contain at most one constrained critical clique, there exists a $O(|V|^3)$ time algorithm to recognize whether G has 5th phylogeny root tree T, and if so, return such a T.*

Proof. Given G, find $CC(G)$ and create the forest of tree chordal graphs $\mathcal{T}$ by decomposing G. Let set S correspond to critical cliques in G of size 1. For each $T_i \in \mathcal{T}$ find the corresponding 3rd Steiner root tree S_i; if one does not exists, by Lemma 3.1.1, return no. For each maximal clique K_i create a Steiner point s_i. For each unconstrained critical clique $c \in K_i$, attach its representative to s_i. By Lemma 5.2.1, each critical clique as $c4$ is contained in at least two large maximal cliques, if not, return no; it follows by the precondition, that each of these maximal cliques will have all other critical cliques each with a single representative. By Lemma 5.2.1, create a Steiner point, s for a critical clique as in $c4$, place s adjacent to the critical clique's representative and to all s_i for each K_i. s will have degree at least three as there is at least two maximal cliques K_i. For a critical clique c in K_i as in $c1a$, where $|c| = 1$, check if c is part of at least two other maximal cliques. For a critical clique c in K_i as in $c1a$ with $|c| > 1$, $c1b$, or $c3$, let s_i be adjacent to the Steiner point adjacent to c's representative. For critical clique in K_i as in $c5$, let s_i be adjacent to the degree one representative.

Our built 3rd Steiner root tree T' is now connected as G was connected and we have connected all the tree chordal graphs by their maximal cliques. As each critical clique had at most one constrained critical clique, each maximal clique will have diameter at most 3 in T'; this satisfies Lemma 4.2.1. As the minimum diameter of a maximal clique in T' is 2 and as $c4$ is the only case where a representative is adjacent to another representative, it follows that all nonadjacent critical clique's representatives are at distance at least 4. All size one representatives in every T_i will all be internal now as we assumed no small leaves exist. Therefore the algorithm, produces an S-restricted 3rd Steiner root tree T'. To produce the 5th phylogeny root tree T, we replace each representative with a Steiner point and place the representatives adjacent to this Steiner point. By Lemma 3.1.1, we have a valid 5th phylogeny root tree T. This construction is $O(|V| \cdot (|V| + |E|)) \in O(|V|^3)$, as it calls ALG(G) at most once for each critical clique and performs a linear amount of work for each of these cliques. □

5.2.2 Structural Restriction 2

In following section, we assume that the input graph G contains no small leaves. A strictly chordal graph may have a large maximal clique having more than one constrained critical clique; if all except one cannot be modified to be unconstrained, then the algorithm returns no, by Corollary 5.1.4. If a critical clique in a 3-root Steiner tree is as $c4$, Lemma 5.2.1 forces the structure for all maximal cliques it is contained in; $c3$ and $c5$ are similarly restrictive.

Lemma 5.2.3. *Let G be a graph with an S-restricted 3rd Steiner root tree T with $S = \emptyset$ and a maximal clique K. If K has a critical clique c as in $c3$ or $c5$ of Lemma 4.2.2 then any other maximal cliques with critical clique cardinality three or more containing C will have all critical cliques as unconstrained.*

Proof. As a representative of the critical clique c is adjacent to a critical clique not in K; any other critical clique that is adjacent to c must be at distance of exactly three from this representative. Therefore, similar to Lemma 5.2.1 assign these critical cliques a single representative. $\square$

Thus, given a representative as in cases $c3$, $c4$, or $c5$, we can immediately decide if the maximal clique can be recombined. As $c2a$ and $c2b$ both have a single representative adjacent to a Steiner point of degree at least three, we now deal with the cases $c1a$ and $c1b$.

Lemma 5.2.4. *Let G be a graph with an S-restricted 3rd Steiner root tree T and c be a critical clique, where c is part of maximal cliques $K_1, K_2, ..., K_n$ and K_1 is as in $c1a$ or $c1b$, then at least one maximal clique must have all critical cliques other than c unconstrained.*

Proof. If all maximal cliques have two constrained critical clique, then at least one maximal clique will have diameter of four. $\square$

Theorem 5.2.5. *Let G be a connected strictly chordal graph, G contains no small leaves, there exists a $O(|V|^3)$ time algorithm to recognize whether G has a 5th phylogeny root tree T, and if so, return such a T.*

Proof. We proceed as Theorem 5.2.2 until we recombine large maximal cliques. For a maximal clique that contains a critical clique as in $c3$, $c4$ or $c5$, by Lemmas 5.2.1 and 5.2.3 we know that all other critical cliques must have a single representative; if not, no 5th phylogeny root will exit. For a critical clique c in K_i as in $c1a$, where $|c| = 1$, apply Operation 1 by searching, if necessary. For each critical clique c as in case $c1a$ with $|c| > 1$ or $c1b$, if it is part of exactly two maximal cliques, then the large maximal cliques containing c must have all its critical cliques unconstrained, by Lemma 5.2.4; if yes, create a Steiner point and set the Steiner point of the $c1b$ critical clique and each of the representatives for each critical clique adjacent to it.

Let set M consist of all maximal cliques containing at least two critical cliques as in case $c1b$ or $c1a$ with $|c| > 1$. Let set N consist of all maximal cliques containing a critical clique as in $c1b$ or $c1a$ with $|c| > 1$ and has all other critical cliques unconstrained. By Lemma 5.2.4, find a critical clique c which is contained in both N and M; perform Operation 1 with searching on this maximal clique and, if possible, remove c from M. Continue until either M is empty, the algorithm has resolved all maximal cliques in M, or $N \cap M$ contains no such c, and therefore, no phylogeny tree will exist.

The algorithm removes one c from the list each time, and we will have at most $O(|V|)$ searches of the maximal cliques, therefore, this will runtime is bounded by $O(|V| \cdot (|V| + |E|)) \in O(|V|^3)$.

When $M = \emptyset$, every maximal clique will only contain critical cliques that are unconstrained. Therefore create a Steiner point for each maximal clique and set each representative adjacent to it. We will now have an S-restricted 3rd Steiner root tree and, thus, the a 5th phylogeny root tree. The construction is $O(|V|^3)$ as we use the polynomial construction from Theorem 5.2.2 and the searching of M takes polynomial time. □

5.2.3 No Restrictions

In any maximal clique there exists at most one leaf critical clique; otherwise, these multiple critical clique would have the same set of neighbors, and therefore, would be a larger critical clique. Similar to Corollary 4.1.2 the following lemma shows a size one leaf critical clique could never exist in a maximal clique of critical clique cardinality two.

Lemma 5.2.6. *Let G be a connected graph with an S-restricted 3rd Steiner root T. If G contains at least three critical cliques then there exist no size 1 leaf nodes in $CC(G)$.*

Proof. Let u be a size 1 leaf node in $CC(G)$. As G is connected, u has exactly one neighbour v in $CC(G)$, and as G contains at least three critical cliques v has at least one neighbour w in $CC(G)$. If v and w are not part of a large maximal clique in G, then by the decomposition of a strictly chordal graph in to tree chordal graphs, it follows that result from Corollary 4.1.2 holds. If v and w are part of a large maximal clique it similarly follows that if u's representative $r(u)$ is internal, then $r(u)$ must be on a path between at least two representatives of v. Thus, the representatives of w are adjacent to $r(u)$. □

As the construction for graph with less than three critical cliques is trivial, for the remainder of the paper let S contain all size one critical cliques in G.

Lemma 5.2.7. *Given a strictly chordal graph G and a corresponding S-restricted 3rd Steiner root tree T, if there exists a small leaf l in a maximal clique K, then:*

1. *l is internal in T,*

2. *each critical clique $c \in K \setminus l$ has all adjacent critical cliques not in K at a distance of at least 2 in T,*

3. *at least one critical clique $c \in K \setminus l$ has all adjacent critical cliques not in K at a distance of 3 in T, and*

4. *every critical clique in K has a single or 2 adjacent representatives.*

Proof. Assume G has such a maximal clique l, by definition of S-restricted 3rd Steiner root tree, it must be internal. As the maximum diameter of a maximal clique is 4 and l is internal, any critical clique not in K adjacent to a critical clique K would be adjacent to l, thus claim two holds. The third claim follows as l is internal and therefore is adjacent to at least one other critical clique c; critical cliques adjacent to c must be at distance of three from c in T, otherwise, adjacent to l. If K has an internal critical clique, c, with at least two nonadjacent representatives in T, l will be adjacent to the same center Steiner point or representative of the critical clique and will be adjacent to all c's neighbors. A critical clique not in K can be adjacent to a critical clique with a single representatives or two adjacent representative and not to l, thus, the fourth claim holds. □

By this lemma, a critical clique c in a maximal clique containing a small leaf can be as in $c1a$, $c1b$, $c2a$ or $c2b$. $c4$ is impossible as the critical clique is adjacent to another critical clique failing to satisfy condition 2. $c3$ is impossible as the small leaf would have to be at a distance of exactly three from single representative, but then it would be a leaf in T, failing to satisfy condition 1. Similarly, $c5$ a small leaf would be distance three from the degree two representative but then a leaf in T.

We now introduce two operations to change a critical clique to satisfy condition 3. The algorithm applies these operations if no suitable critical clique exists to satisfy condition 3 of Lemma 5.2.7. Notice that only one of these operations can apply to a set of maximal cliques. In addition, the search as done for Operation 1 can be applied to these operations.

Definition 5.2.2 (Operation 2a). *Given a critical clique, c, which is part of at least three maximal cliques, $K_1, K_2, ..., K_n$. If at most one of $K_2, ..., K_n$ was part of a decomposed tree chordal graph with c as in $c1a$, $c2b$ $c2a$, or $c2b$ and all critical cliques in the remainder are unconstrained. Then give all critical cliques c a single representative and let the Steiner point adjacent to c be adjacent to the Steiner points from $K_2, ..., K_n$ such that each remaining critical clique is at distance exactly three from c.*

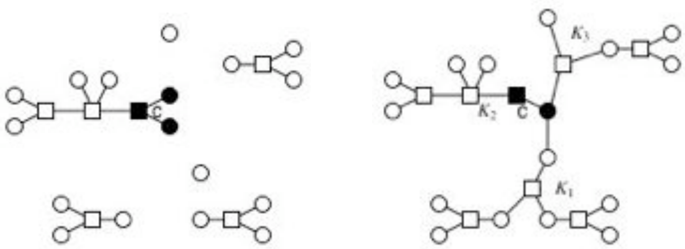
Definition 5.2.3 (Operation 2b). *Given a critical clique, $|c| \geq 2$, which is part of exactly two large maximal cliques K_1 and K_2. If all critical cliques in K_1 and K_2 other than c are unconstrained then create two Steiner points, p_1 and p_2; let all critical cliques in K_1 other than c be adjacent to p_1, all critical cliques in K_2 other than c be adjacent to p_2, and give c two adjacent representatives where one is adjacent to p_1 and the other to p_2.*

Lemma 5.2.8. *Given a graph with an S-restricted 3rd Steiner root tree and a large maximal clique K containing a small leaf l, then one critical clique $C \in K \setminus l$ can have Operation 2a applied, Operation 2b applied, or is as $c2a$.*

Proof. Lemma 5.2.7 case 3 shows that at least one critical clique c must have all critical cliques adjacent to c not in K at distance exactly 3, the above enumerates the possibilities. To see that there

(a) CC(G) with critical cliques labelled by size.

(b) Forest of tree chordal graphs from decomposition of $CC(G)$.



(c) S-restricted 3rd Steiner root trees for tree chordal graphs in (b).

(d) S-restricted 3rd Steiner root tree for G after Operation 2a.

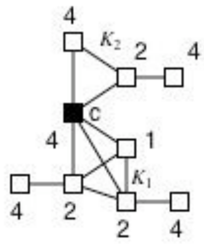
Figure 5.1: An example of Operation 2a for a strictly chordal graph G.

exists no other situations to consider, we note that cases $c1a$, $c1b$, $c2a$, $c2b$ are the only cases for a maximal clique with a small leaf. Operations 2a and 2b show how to construct such a distance 3 situation. For a critical clique c as in $c2b$, representing the critical clique by two adjacent representatives will force all critical cliques in K to have all representatives in their maximal cliques at distance 3 from c's representative; thus, at least one can have Operation 2a applied, Operation 2b applied, or is as $c2a$. $\square$

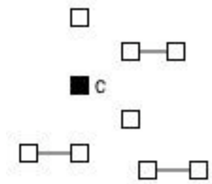
Lemma 5.2.9. *Given a strictly chordal graph G and a corresponding S-restricted 3rd Steiner root tree T, if there exists a small leaf l in a maximal clique K, then:*

1. *if $\text{cccard}(K) = 3$ and there exists exactly one critical clique $c \in K \setminus l$ with two adjacent representative, then all other critical cliques have all adjacent critical cliques not in K at a distance of exactly 3 in T,*

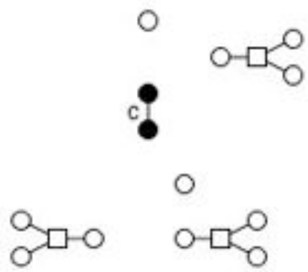
2. *if $\text{cccard}(K) = 3$ and no critical clique $c \in K \setminus l$ has two adjacent representative, then all critical cliques having all adjacent critical cliques not in K at a distance of exactly 3 in T, and*



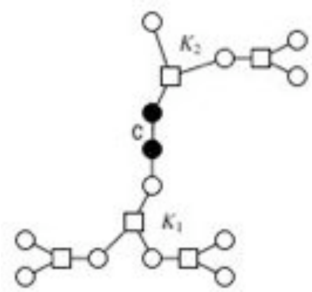
(a) CC(G) with critical cliques labelled by size.



(b) Forest of tree chordal graphs from decomposition of $CC(G)$.



(c) S-restricted 3rd Steiner root trees for tree chordal graphs in (b).



(d) S-restricted 3rd Steiner root tree for G after Operation 2b.

Figure 5.2: An example of Operation 2b for a strictly chordal graph G.

3. *if* $\text{cccard}(K) \geq 4$ *then there exists a critical clique* $c \in K \setminus l$ *with Operation 2a applicable or* c *is as* $c2a$.

Proof. When $cccard(K) = 3$ and l is internal, at least one of the critical cliques must be adjacent to l, as it is internal. If the other critical clique is not adjacent and does not have two adjacent representatives then the Steiner point adjacent to l will have degree two, a contradiction. Thus, the first claim holds. In case 2, as every critical clique has two adjacent representative or a single representative, the two non-leaf critical cliques have single representatives that are adjacent to the small leaf, otherwise, the maximal clique has width four. It follows from Lemma 4.2.2, that Operation 2a and $c2a$ represent the only cases for the critical clique c in its other maximal cliques. When $cccard(K) \geq 4$, at least one critical clique representative r will have a single representative adjacent to l, otherwise, not internal. All critical cliques adjacent to r that are not in not in K must be at distance of three, otherwise, adjacent to r. Therefore, the third case holds.

□

Theorem 5.2.10. *Let G be a strictly chordal graph. Then there exists a $O(|V|^3)$ time algorithm to recognize whether G has a 5th phylogeny root tree T, and if so, return such a T.*

Proof. Proceed as in Theorem 5.2.2 until the recombination of large maximal cliques. Lemmas 5.2.1 and 5.2.3 handle tree chordal graphs returned as in $c3$, $c4$, and $c5$. Define set L as all maximal cliques K which contain a small leaf l. All critical cliques as in $c1a$ and $c1b$ which are in a maximal clique in L will need to be changed using either Operation 1 or 2a with searching. For each $K \in L$ such that $cccard(K) = 3$ if one critical clique c in K is as $c2b$ and in no other maximal clique, then if $|c| > 1$ then given c two adjacent representatives such that one is adjacent to l. All other critical cliques must be one of the choices in Lemma 5.2.8, otherwise return no. If $|c| = 1$ then no S-restricted 3-root Steiner tree will exist by Lemma 5.2.9. Otherwise by Lemma 5.2.9 both critical cliques must be one of the choices in Lemma 5.2.8, otherwise return no. Perform operations if needed and set representatives adjacent to l.

All maximal cliques in L now have critical clique cardinality at least 4. If any critical clique is as in $c2a$ and contained in exactly two maximal cliques, then set the leaf adjacent to it and all remaining critical cliques single representative adjacent to a Steiner point adjacent to the leaf. For a maximal clique containing multiple critical cliques as in $c1a$ or $c1b$, first apply Operation 2a, if possible, and then, apply Operation 1 if possible. If neither operation is applicable, then no tree exists by Lemma 5.2.7. Every maximal clique in L must have a critical clique changed by Operation 2a, otherwise by Lemma 5.2.9 no S-restricted 3-root Steiner tree exists; combine these maximal cliques by setting l adjacent to this critical clique. Set the other critical cliques, which are now all necessarily unconstrained, adjacent to a Steiner point adjacent to l. All maximal cliques in L will now be recombined, by Lemma 5.2.9.

Finally continue as in Theorem 5.2.5 by combining large maximal cliques with more than one constrained critical clique. We produce a 5th phylogeny root tree as in Theorem 5.2.2. This construction adds a linear amount of work for each maximal clique containing a small leaf. Therefore, the algorithms overall runtime, as in Theorem 5.2.5, is still $O(|V|^3)$.

□

Chapter 6

Conclusions and Future Research

In this section we summarize our major results and give some open problems.

For Chapter 1 the result by [24] that all strictly chordal graphs have a kth leaf root tree if $k \geq 4$ leads naturally to the following question.

Problem 6.0.1. *Characterize those graphs which are kth leaf powers, for $k \geq 3$.*

For strictly chordal graphs, there exists a linear time algorithm to compute a leaf root and a linear time algorithm to construct a 3rd Steiner root tree. The following problem is open.

Problem 6.0.2. *Let G be a chordal graph and let k be an integer such that $k \geq 3$. Either give a polynomial (preferably linear) time algorithm to decide if a kth root Steiner tree T exists for G, and if so construct T, or show k-SRP $\in$ NP-complete.*

In Chapter 2 we introduce strictly chordal graphs, a subclass of chordal graphs, for which structure properties allow efficient solutions to be developed for all three leaf-labeled root problems. We introduce and characterize this class of graphs.

In Chapter 3 we describe the S-restricted kth Steiner root problem and show its equivalence to the $(k+2)$th phylogenetic root problem.

Problem	Known Results	Open Problems
k-PRP	$k \leq 4$, $O(\|V\| + \|E\|)$ solution [27] $k = 5$, tree chordal graphs, $O(\|V\| + \|E\|)$ solution* $k = 5$, strictly chordal graphs, $O(\|V\| + \|E\|)$ solution* $k \geq 5$, bounded degree in tree, $O(\|V\| + \|E\|)$ solution [8]	$k \geq 5$, unknown
k-SRP	$k \leq 2$, $O(\|V\| + \|E\|)$ solution [27] $k = 3$, strictly chordal graphs, $O(\|V\| + \|E\|)$ solution [24]	$k \geq 3$, unknown
k-LRP	$k \leq 4$, $O(\|V\| + \|E\|)$ solution [32] $k \geq 4$, strictly chordal graphs, $O(\|V\| + \|E\|)$ solution*	$k \geq 5$, unknown
kth tree root	$k \geq 1$, $O(\|V\|^3)$ solution [29, 22]	
kth root	$k = 2$, NP-complete [31]	$k \geq 3$, unknown

Figure 6.1: A summary of the best known results for various root construction problems. Problems marked by a '*' are considered in this thesis.

In Chapter 4, we derive an algorithm to decide if a tree chordal graph has a 5th phylogenetic root tree, and if so, construct such a root. We present the class of tree chordal graphs as an intermediate step for the final construction in Chapter 5.

Problem 6.0.3. *Let G be a tree chordal graph and let k be an integer such that $k \geq 6$. Give a polynomial (preferably linear) time algorithm to decide if a kth root phylogenetic tree T exists for G, and if so construct T.*

In Chapter 5, we present a polynomial time algorithm to construct a 5th phylogenetic root tree for a strictly chordal graph if one exists. This is the largest class of graphs for which a polynomial time algorithm for k-PRP such that $k \geq 5$ is known.

Problem 6.0.4. *Let G be a chordal graph and let k be an integer such that $k \geq 5$. Either give a polynomial (preferably linear) time algorithm to decide if a kth root phylogenetic tree T exists for G, and if so construct T, or show k-PRP $\in$ NP-complete.*

Bibliography

[1] A. D. Baxevanis and B. F. F. Ouellette. *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*. John Wiley and Sons, Inc., second edition, 2001.

[2] C. Berge. Some classes of perfect graphs. In *Six Papers on Graph Theory*, pages 1–21. Indian Statistical Institute, Calcutta, 1963.

[3] C. Berge. *Graphs and Hypergraphs*. North-Holland Publishing Company, 1973.

[4] A. Berry and J. P. Bordat. Moplex elimination orderings. *Preprint submitted to Elsevier Science*, 2001.

[5] A. Brandstadt, F. Dragan, C. Chepoi, and V. Voloshin. Dually chordal graphs. *SIAM Journal on Discrete Mathematics*, 11(3):437–455, 1998.

[6] A. Brandstadt, V. B. Le, and J. P. Spinrad. *Graph Classes: A Survey*. SIAM, Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, 1999.

[7] P. Buneman. A characterization of rigid circuit graphs. *Discrete Mathematics*, 9:205–212, 1974.

[8] Z. Chen, T. Jiang, and D. G. Corneil. Computing phylogenetic roots with bounded degrees and errors. *SIAM Journal of Computing*, 32(4):864 – 879, 2003.

[9] M. Chudnovsky, P. Seymour, N. Robertson, and R. Thomas. The strong perfect graph theorem. *Annals of Mathematics* to appear, 2002.

[10] V. Chvatal, J. Fonlupt, L. Sun, and A. Zemirline. Recognizing dart-free perfect graphs. *SIAM Journal of Computing*, 31(5):1315–1338, 2002.

[11] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, Cambridge, Cambridge, Massachusetts, second edition, 2001.

[12] W. H. E. Day. Computational complexity of inferring phylogenies from dissimilarity matrices. *Bulletin of Mathematical Biology*, 49:461–467, 1987.

[13] G. Dirac. On rigid circuit graphs. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 25:71–76, 1961.

[14] A. W. F. Edwards and L. L. Cavalli-Sforza. Reconstruction of evoultionary trees. In V. H. Heywood and J. McNeill, editors, *Phenetic and phylogenetic classification. Systematics Association*, pages 67–76, London, 1964.

[15] R. W. Floyd. Algorithm 97: Shortest path. *Communications of the Association for Computing Machinery*, 5(6):345, 1962.

[16] S. Freeman and J. C. Herron. *Evolutionary Analysis*. Prentice-Hall, Inc., 1998.

[17] D. R. Fulkerson and O. A. Gross. Incidence matrices and interval graphs. *Pacific Journal of Mathematics*, 15:835–855, 1965.

[18] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman and Company, 1991.

[19] F. Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *Journal of Combinatorial Theory, Series B*, 16:47–56, 1974.

[20] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, Inc., 1980.

[21] R. Haenni and N. Lehmann. Efficient hypertree construction. Technical Report 99-3, Institute of Informatics, University of Fribourg, 1999.

[22] P. E. Kearney and D. G. Corneil. Tree powers. *Journal of Algorithms*, 29(1):111 – 131, 1998.

[23] W. Kennedy and G. Lin. 5th phylogenetic root construction for strictly chordal graphs. submitted to ISAAC, 2005.

[24] W. Kennedy, G. Lin, and G. Yan. Strictly chordal graphs are leaf powers. *Journal of Discrete Algorithms*, 2005.

[25] C. Lekkerkerker and D. Boland. Representation of finite graphs by a set of intervals on the real line. *Fundamenta Mathematicae*, 51:45–64, 1962.

[26] W. Li. *Molecular Evolution*. Sinauer Associates, Inc., 1997.

[27] G. Lin, T. Jiang, and P. E. Kearney. Phylogenetic k-root and steiner k-root. In *The 11th Annual International Symposium on Algorithms and Computation (ISAAC 2000)*, volume 1969, pages 539–551, 2000.

[28] G. Lin, W. Kennedy, H. Kong, and G. Yan. The 5-root phylogeny construction for tree chordal graphs. submitted to Journal of Discrete Applied Mathematics, 2005.

[29] Y. L. Lin and S. S. Skiena. Algorithms for square roots of graphs. *SIAM Journal on Discrete Mathematics*, 8:99 – 118, 1995.

[30] W. P. Maddison, M. J. Donoghue, and D. R. Maddison. Outgroup analysis and parsimony. *Systematic Zoology*, 33:83 – 103, 1984.

[31] R. Motwani and M. Sudan. Computing roots of graphs is hard. *Discrete Applied Mathematics*, 54:81–88, 1994.

[32] N. Nishimura, P. Ragde, and D. M. Thilikos. On graph powers for leaf-labeled trees. In *Proceedings of the 7th Scandinavian Workshop on Algorithm Theory (SWAT 2000)*, volume LNCS 1851, pages 125–138, 2000.

[33] Pavel A. Pevzner. *Computational Molecular Biology: An Algorithmic Approach*. Massachusetts Institute of Technology, 2000.

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

[35] N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4:406–425, 1987.

[36] S. Sattath and A. Tversky. Additive similarity trees. *Psychometrika*, 42:319–345, 1977.

[37] J. P. Spinrad. *Efficient Graph Representations*. AMS, 2003.

[38] L. Sun. Two classes of perfect graphs. *Journal of Combinatorial Theory, Series B*, 53:273–291, 1991.

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

[40] S. Warshall. A theorem on boolean matrices. *Jounal of the Association for Computing Machinery*, 9(1):11–12, 1962.

[41] D. B. West. *Introduction to Graph Theory*. Prentice Hall, second edition, 2001.

[42] R. Wilson. *Introduction to Graph Theory*. Pearson Education Limited, fourth edition, 1996.