Spring 5-17-2016

Tree Graphs and Orthogonal Spanning Tree Decompositions

James Raymond Mahoney
Portland State University

Recommended Citation

Mahoney, James Raymond, "Tree Graphs and Orthogonal Spanning Tree Decompositions" (2016). *Dissertations and Theses.* Paper 2944.

10.15760/etd.2939

Tree Graphs and

Orthogonal Spanning Tree Decompositions

by

James Raymond Mahoney

A dissertation submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy
in
Mathematical Sciences

Dissertation Committee:
John Caughman, Chair
Nirupama Bulusu
Derek Garton
Paul Latiolais
Joyce O'Halloran

Portland State University
2016

Abstract

Given a graph G, we construct $\mathcal{T}(G)$, called the *tree graph* of G. The vertices of $\mathcal{T}(G)$ are the spanning trees of G, with edges between vertices when their respective spanning trees differ only by a single edge. In this paper we detail many new results concerning tree graphs, involving topics such as clique decomposition, planarity, and automorphism groups. We also investigate and present a number of new results on orthogonal tree decompositions of complete graphs.

Dedication

This dissertation is dedicated to Brigette, who cheerfully humors me whenever I inevitably start speaking math to her. Also to my family, whose support and encouragement have helped me focus and go further in my education than I ever thought possible.

Acknowledgements

First and foremost I need to thank John Caughman for all of his time and effort as my adviser during my years at Portland State. John introduced me to graph theory and took a chance by becoming my Master's adviser without knowing much about me. His tireless dedication to all of his students is remarkable. John's suggestions for topics of inquiry and methods of attack made much of this work possible, while his keen editing eye greatly improved the quality of its presentation. I cannot thank you enough.

Thank you to the rest of my committee members for believing that my work was worthwhile.

Thank you to the math department at Portland State University for providing me with the Enneking Fellowship for two years. This generous support allowed me to spend my academic hours researching instead of teaching.

Thank you to all of my professors at Portland State University and the University of Portland for helping me realize and achieve my educational goals.

Thank you to Gary Gordon, who introduced me to matroid basis graphs and helped me discover the connection between my original research and tree graphs.

Thank you to Georgia Reh for assistance in editing this paper.

Lastly, thank you to Brendan McKay for the *nauty* program, to Ed Scheinerman for his *matgraph* MATLAB package, and to the development group behind the *Geometer's Sketchpad* software, all of which were critical to my visualization and understanding of tree graphs.

Table of Contents

List of Figures

Glossary of Symbols

$\alpha(G)$	Independence number of graph G
$\chi(G)$	Chromatic number of graph G
$\delta(G)$	Minimum degree of graph G
$\Delta(G)$	Maximum degree of graph G
$\kappa(G)$	Vertex connectivity of graph G
$\kappa'(G)$	Edge connectivity of graph G
$\mu(G)$	Number of spanning unicycles in graph G
$\tau(G)$	Number of spanning trees of graph G
$\theta_{a,b,c}$	Theta graph
$A(G)$	Adjacency matrix of graph G
$\mathrm{Aut}(G)$	Automorphism group of graph G
$C(G)$	Center of graph G
$\mathrm{circ}(G)$	Circumference of graph G
C_n	Cycle graph on n vertices
$\mathrm{diam}(G)$	Diameter of graph G
$\mathrm{d}(x, y)$	Distance between vertices
D_n	Dihedral group with n elements
$\mathrm{ecc}(G)$	Eccentricity of graph G
$E(G)$	Edge set of graph G
f	Number of faces in a given plane graph
$\mathcal{F}$	1-factorization of K_{2n}
$g(G)$	Glory of graph G
$\overline{G}$	Complement of graph G
G^*	Planar dual of graph G

$G \pm e$	Edge e added to/deleted from graph G
$G \cdot e$	Edge e contracted in graph G
K_n	Complete graph on n vertices
$K_{r_1,r_2,\ldots,r_k}$	Complete multipartite graph
$L(G)$	Line graph of graph G
m	Number of edges of a given graph
M_G	Graphic matroid of graph G
n	Number of vertices of a given graph
$OPG_t(\cdot)$	Opposing pair graph for K_{2n}
$P_{n,k}$	Graph joining two vertices with n openly-disjoint paths of length k
$\mathrm{rad}(G)$	Radius of graph G
S_n	Symmetric group on n elements
$SG(\cdot)$	Starter graph for K_{2n}
$\mathcal{T}(G)$	Tree graph of graph G
$v(G)$	Equal to $m - n + 1$ in graph G
$V(G)$	Vertex set of graph G
W_n	Wheel graph on n vertices
$\mathbb{Z}_n$	Cyclic group on n elements
$\sim$	Adjacency
$\binom{a}{b}$	Binomial coefficient
$\vert\ \vert$	Cardinality
$\square$	Cartesian product of graphs
$\times$	Cartesian product, direct product
$\cap$	Intersection
$\cong$	Isomorphic

$\sqsubseteq$	Minor
$\leq$	Subgraph, subgroup
$\subseteq$	Subset
Δ	Symmetric difference
$\cup$	Union
$\bigodot$	Vertex identification operation

1 Introduction

1.1 Overview

Given a graph G, we can construct a new graph $\mathcal{T}(G)$, called the *tree graph* of G. The vertices of $\mathcal{T}(G)$ are the spanning trees of G. We place an edge between vertices x and y in $\mathcal{T}(G)$ when their respective spanning trees differ only by a single edge.

Tree graphs have been studied since at least 1966, when Cummins [4] wrote an influential paper defining tree graphs and showing that they are hamiltonian. Although this original paper was published in an IEEE engineering journal, graph theorists soon took interest in the objects and additional papers found their way into mathematics journals. Some were motivated to study tree graphs because of their possible use as topologies for networks [26].

The first wave of tree graph results emerged from the late 1960s through the mid 1970s. These papers continued to investigate properties of tree graphs relating to hamiltonicity [10, 13] as well as the connection between tree graphs and the broader category of mathematical objects called matroid basis graphs [20, 21].

Interest in tree graphs was renewed by additional papers published from the late 1980s through the early 2000s. This second wave of research concerned itself with graph properties such as connectivity [1, 16] and chromatic number [26]. Variants of tree graphs were also considered during this time [24, 25].

While many papers about tree graphs have been published over the last half-century, there is still much that remains to be considered, including many classic graph parameters, such as the clique number and independence number. Other standard graph theoretic concepts such as planarity, decomposition, and regularity, as well as more algebraic topics, like automorphism groups, vertex-transitivity, and integrality, are likewise ripe for exploration.

When a new mathematical object is defined, many questions immediately arise. What properties does this object have? What is the relationship between this object and other known objects? In this way, research into tree graphs is similar to research into line graphs. Given a graph G, there is a deterministic construction (given in Section 4.8) one can follow to produce its line graph, $L(G)$. Line graphs have been defined since at least the early 1930s, and have proven to be a very fruitful area of graph-theoretic research. We can answer questions about relationships between properties of a graph and properties of its line graph. We can predict the structure of line graphs and determine when a graph is a line graph. These same lines of inquiry can be extended to tree graphs, and, as mentioned earlier, many important questions are still open.

This dissertation aims to make a significant contribution to the study of tree graphs by filling in many of the missing pieces in our understanding of these objects. We have several novel results and many conjectures that are likely to lead to further proofs. For example, we have discovered a structural property of tree graphs that may lead toward a characterization, based on their relationship to matroid basis graphs. We have also found an infinite family of integral tree graphs, which is of importance to graph theory even outside of the realm of tree graphs. This research will help paint a more accurate picture of an object that has already been given considerable, and well-deserved, attention.

1.2 Organization

Chapter 2 provides a background on spanning trees and tree graphs, with definitions, examples, and preliminary results. The heart of this paper is broken down into three main explorations: Chapter 3 will look at the tree graph construction as a function from the set of graphs to itself. Here we consider questions of injectivity, surjectivity, fixed points, and other functional properties. We also investigate relationships found between graphs and their tree graphs. Specifically, in what ways can we relate parameters of graphs to parameters of their tree graphs? In Chapter 4 we will consider tree graphs as a family and describe properties that they all share. We will attempt to provide a classification of tree graphs, and discuss special classes, such as regular tree graphs. In Chapter 5, we investigate spanning tree decompositions of complete graphs. In particular, we explore the Brualdi-Hollingsworth conjecture which states that every 1-factorization of K_{2n} has a full set of n disjoint orthogonal spanning trees. Appendix A contains some of the algorithms used in the course of this research. In Appendix B we illustrate many of the common graph families which appear in this paper. Appendix C is a collection of spectrum data for a particular family of tree graphs. Appendix D is a partial catalog of data collected on tree graphs generated for this research, while Appendix E contains data relevant to specific 1-factorizations from the last chapter. Each chapter will contain relevant previous work from the literature as well as novel results. Many conjectures and open problems will be given as well.

2 Background

2.1 Graph Terminology

A *graph* $G = (V, E)$ is a set V of *vertices* together with a multiset E of *edges*, which is made up of unordered pairs of vertices from V. Edges are said to connect vertices, and if two vertices x and y are connected by an edge they are said to be *adjacent*, which we denote by $x \sim y$. We can also refer to the edge xy. The *degree* of a vertex is the number of edges incident to it. If there is an edge from a vertex to itself, that edge is called a *loop*. If there are no repeated edges or loops in E, G is called a *simple* graph. Unless stated otherwise, all graphs discussed in this paper will be assumed to be simple and have a finite number of vertices.

The *complete graph* K_n has n vertices, all of which are adjacent to each other. Complete graphs are also called *cliques*. A 3-clique is sometimes called a *triangle*. *Complete multipartite graphs*, such as $K_{a,b}$ and $K_{a,b,c}$, partition their vertices into the sizes given by their subscripts, where two vertices are adjacent if and only if they are in different cells of the partition. A *path* of length n is an alternating sequence of vertices and edges $v_1, e_1, v_2, e_2, \ldots, v_n, e_n, v_{n+1}$ such that each edge e_i joins v_i and v_{i+1} and no two vertices are the same. A *cycle* of length n is a path of length n, but where the first and last vertices are the same. Cycles must contain at least three edges. Figure 1 shows examples of some of these graphs.

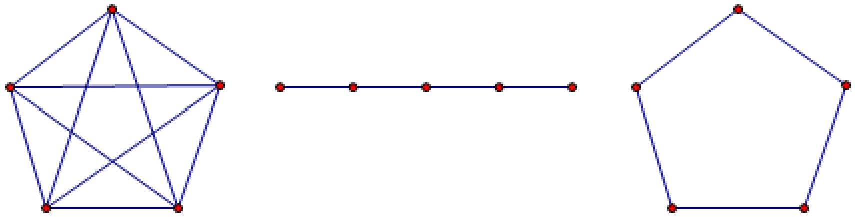


Figure 1: Depictions of K_5, P_5, and C_5

A graph with no cycles is called a *forest*. If there is a path between every pair of vertices in a graph, the graph is said to be *connected*. A *tree* is a connected forest. The vertices of degree one in a graph are called *pendant* vertices, and the pendant vertices in a tree are called *leaves*. A *subgraph* $H = (V', E')$ of a graph G is any graph where $V' \subseteq V$ and $E' \subseteq E$. This relation is denoted by $H \leq G$. A maximally connected subgraph of a graph is called a *component*. A vertex (edge) whose deletion from a graph increases the number of components is called a *cut vertex (cut edge)*. A graph is said to be *k-connected* if at least k vertices need to be removed in order to disconnect it. The complete graph K_n is said to be k-connected for all $k < n$. A *spanning tree* of a connected graph G is a subgraph that contains all of the same vertices as G and is a tree.

Two graphs G and H are *isomorphic*, written $G \cong H$, if they have exactly the same structure. Specifically, graphs are shown to be isomorphic by finding a bijection between their vertices that preserves the adjacency relationship, i.e., an isomorphism.

Throughout Chapter 2, we assume graphs are connected unless otherwise stated. In general, we will reserve the parameters $m = |E(G)|$ and $n = |V(G)|$ for the number of edges and vertices, respectively, in a graph.

2.2 Preliminary Results on Spanning Trees

The following elementary results will be used repeatedly in later sections of the paper.

Lemma 2.1. *[32, Theorem 2.1.4] Every spanning tree of a graph with n vertices has $n - 1$ edges.*

Lemma 2.2. *[32, Corollary 2.1.5.c] Every graph has a spanning tree.*

Lemma 2.3. *[32, Propositions 2.1.6-7] If T and T' are spanning trees of a graph G and $e \in E(T) - E(T')$, then there is an edge $e' \in E(T') - E(T)$ such that $T - e + e'$ and $T + e' - e$ are both spanning trees of G.*

Lemma 2.4. *For any spanning tree T of a graph G and any edge $e \in E(G)$, there exists an edge $e' \in E(T)$ such that $T + e - e'$ is a spanning tree of G.*

Proof. If $e \in E(T)$, let $e' = e$. Otherwise, adding e to T produces exactly one cycle (see Corollary 2.1.5.b in [32]). Remove any other edge e' of that cycle to get back to a spanning tree. □

Lemma 2.5. *Every edge in a graph G is contained in some spanning tree of G.*

Proof. Let $e \in E(G)$ and T be a spanning tree of G that does not contain e. Then by Lemma 2.4, $T + e - e'$ is a spanning tree of G for some edge $e' \in T$. □

Lemma 2.6. *Every acyclic subgraph of a graph G is contained in some spanning tree of G.*

Proof. Let $H \leq G$ be acyclic, and T be a spanning tree for G that minimizes $t = |E(H) - E(T)|$. If $t = 0$, choose T. Now suppose $t \neq 0$. Pick $e \in E(H) - E(T)$. Then $T + e$ has a cycle C. Since H is acyclic, there exists $e' \in E(C) - E(H)$. Then $T + e - e'$ is a spanning tree that contradicts the minimality of t. Therefore $t = 0$ and such a spanning tree exists. □

2.3 Tree Graphs

Graphs are used as an abstract model of relationships between objects, where vertices represent the objects and edges denote relationships between them. In a *tree graph* $\mathcal{T}(G)$, the vertices represent all of the spanning trees of G and the edge relationship describes how near the spanning trees are to each other, in terms of their edge sets. In later sections of this paper, we will often simultaneously treat such a vertex both as a vertex of $\mathcal{T}(G)$ and as a spanning tree of G. When there is no room for confusion, we will sometimes refer to spanning trees simply as trees. Note also that in this paper, every graph that will be used to construct a tree graph will be assumed to be connected. This restriction will be explained at the end of Section 3.1.

Given a graph G, its tree graph $\mathcal{T}(G)$ is constructed in the following way. The vertices of $\mathcal{T}(G)$ are all of the spanning trees of G. Two distinct spanning trees are adjacent in $\mathcal{T}(G)$ if we can get from one to the other by swapping a single edge. This is called the *edge exchange property.* More formally, two trees are adjacent if the size of the symmetric difference of their edge sets is two.

Let us look at this construction for the four-cycle C_4 in Figure 2.

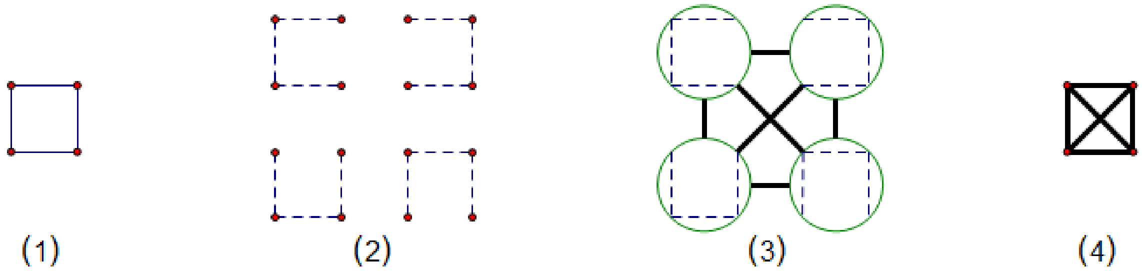


Figure 2: Showing the construction of $\mathcal{T}(C_4)$

First in (1) we see C_4. Then in (2) we see the four spanning trees of C_4 with dashed edges. In (3), we view these four trees as four vertices in a graph with the edge exchange condition for adjacency. Denote the four edges of C_4 by the names T, R, B, and L, for top, right, bottom, and left, respectively. Let us consider which trees the upper left tree should be adjacent to. We can take out L and replace it by R to create the top right tree. We can swap U for R to get the bottom left tree. We can also exchange B for R to get the bottom right tree. In this way, the top left tree is adjacent to every other tree in $\mathcal{T}(C_4)$, so we see the bold edges connecting them. The same is true for every tree in the graph. Thus in (4) we see that the tree graph of the four-cycle is the complete graph on four vertices, or using graph notation, $\mathcal{T}(C_4) \cong K_4$.

Let us consider one more example. This time we will add another edge to C_4 to get $K_4 - e$. Figure 3 shows its tree graph construction. In (1) we see $K_4 - e$. Next in (2) using dashed edges we see its eight spanning trees. The new diagonal edge has given us four more trees to play with. Finally, in (3) in bold we see $\mathcal{T}(K_4 - e)$. By adding one edge to our starting graph we have picked up four more vertices and twelve more edges in the tree graph.

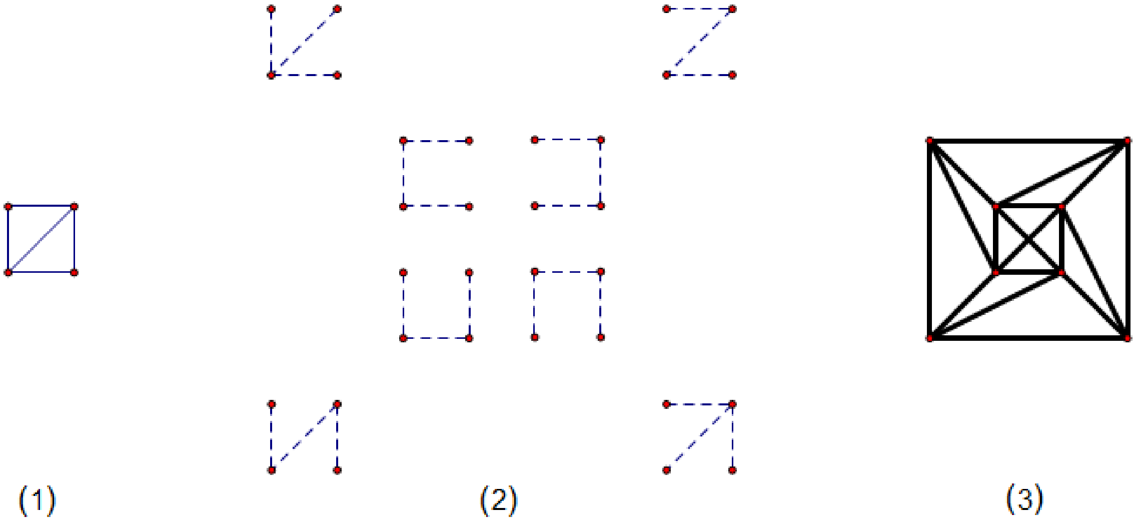


Figure 3: Showing the construction of $\mathcal{T}(K_4 - e)$

Note that our spanning trees are considered different if they have different sets of edges, not solely if they are nonisomorphic. So while $K_4 - e$ has two isomorphism classes of trees, P_3 and $K_{1,3}$, it has eight distinctly labeled trees. These are what we are interested in comparing.

The new results in this paper originated in the generation and study of many examples of tree graphs. Multiple parameters were measured, recorded, and compared to existing data for patterns. The production and analysis of such a data set is a distinguishing feature of this work, and a sizable catalog is included in Appendix D to aid in future investigations.

3 The Tree Graph Function and Parameters

Often when a new graph is constructed from an old one, properties of the new graph can be inferred from properties of the old. These properties may be parametric or structural. For example, the number of vertices in the line graph $L(G)$ of G is the number of edges in G. This is a parametric relation, since we are describing a graph parameter in terms of other known values. We also know that a vertex of degree k in G leads to a k-clique in $L(G)$; this is a structural relation.

We can think of the tree graph construction as a process where we input a graph and get its tree graph as output. This section contains results relating properties of graphs to properties of their tree graphs. Throughout Chapter 3, we assume graphs are connected unless otherwise stated.

3.1 The Tree Graph Function

Following an edge in a tree graph $\mathcal{T}(G)$ amounts to changing one spanning tree of G into another using the edge exchange property. One might wonder, then, if it is always possible to transform one tree into any other by iteration of this process. The following result shows us that this is the case.

Lemma 3.1. *The tree graph $\mathcal{T}(G)$ is connected for all graphs G.*

Proof. Let T and T' be two trees of G. Using Lemma 2.3 repeatedly, we can add edges from $E(T') - E(T)$ while removing edges from $E(T) - E(T')$ until we have swapped in all of the missing edges. This induces a path from T to T' in $\mathcal{T}(G)$. Thus there is a path between every two vertices in $\mathcal{T}(G)$, so it is connected. □

The *distance* between vertices x and y in a graph G is the minimum number of edges in a path from x to y, and is denoted by $\mathrm{d}(x, y)$. Distance between vertices in

$\mathcal{T}(G)$ depends on how similar the corresponding trees are, which we measure by how many edges they have in common. More formally, we have the following.

Lemma 3.2. *For trees T_1 and T_2 in $\mathcal{T}(G)$,*

$$\mathrm{d}(T_1, T_2) = n - 1 - |E(T_1) \cap E(T_2)| = |E(T_1)\Delta E(T_2)|/2$$

where Δ denotes the symmetric difference of sets.

Proof. Every tree has $n-1$ edges. To get from T_1 to T_2, we have to swap the edges that they do not share, of which there are $n-1-|E(T_1)\cap E(T_2)|$. The set of swapped edges occur in pairs in the symmetric difference. Repeated use of Lemma 2.3 swaps these pairs, reducing $t = |E(T_1)\Delta E(T_2)|$ to zero in $t/2$ steps. □

The *eccentricity* of a vertex x, written $\mathrm{ecc}(x)$, is defined by

$$\mathrm{ecc}(x) = \max\{\mathrm{d}(x, y) \mid y \in V(G)\}$$

and represents the farthest away that a vertex can be from x. The minimum eccentricity over all vertices of a graph is called the *radius*, denoted $\mathrm{rad}(G)$. The maximum eccentricity is called the *diameter*, denoted $\mathrm{diam}(G)$. The collection of all vertices of minimum eccentricity is called the *center* of G, and is denoted by $C(G)$.

If two trees have no edges in common, by Lemma 3.2 they would be at distance $n-1$. On the other hand, we have at most $m-(n-1) = m-n+1$ available edges to swap in order to change between two trees. Thus we have the following result.

Lemma 3.3. *For any graph G,*

$$\mathrm{diam}(\mathcal{T}(G)) \leq \min\{n-1, m-n+1\}.$$

In general we cannot say much about the relationship between the diameters of G and $\mathcal{T}(G)$. For example, consider $G \cong K_n$. The diameter of complete graphs is 1, whereas once $n \geq 4$, we can always find two spanning trees that have no edges in common, making the diameter for the tree graphs at least $n-1$. Thus these values can be arbitrarily far apart.

We can view the tree graph operation as a function from the set of connected graphs, $\mathcal{G}$, to itself, i.e. $\mathcal{T} : \mathcal{G} \to \mathcal{G}$. We can thus investigate properties of this function, such as surjectivity, injectivity, pre-images, and fixed points.

Theorem 3.4. *[13, Lemma 1] For any graph G that contains a cycle, $\mathcal{T}(G) \cong G$ if and only if $G \cong K_3$.*

This tells us that the only nontrivial fixed point of the tree graph function is the triangle. We trivially have that $\mathcal{T}(K_1) \cong K_1$.

Theorem 3.5. *[13, Lemma 2] For any $n > 3$, the cycle graph C_n is not a tree graph.*

Roughly, this result says that tree graphs contain much more structure than a simple cycle. It also tells us that the tree graph function is not surjective.

Theorem 3.6. *[13, Theorem 1] Let G be a graph that contains a cycle. The graphs in the iterated tree graph sequence*

$$G, \mathcal{T}(G), \mathcal{T}(\mathcal{T}(G)), \ldots$$

will continue to get larger, either in number of edges or vertices or both, unless $G \cong K_3$.

Thus except in the trivial case, tree graphs are larger than their input graphs.

Lemma 3.7. *The tree graph of a tree is a single vertex.*

Proof. The only spanning tree of a tree T is the tree T itself, so in this case, the tree graph will be K_1. □

Lemma 3.8. *For any $n \geq 3, \mathcal{T}(C_n) \cong K_n$.*

Proof. The cycle C_n has n vertices and n edges. Thus $\operatorname{diam}(\mathcal{T}(C_n)) \leq 1$ by Lemma 3.3. The diameter cannot be zero, since $\mathcal{T}(C_n)$ has $n \geq 3$ vertices by Lemma 3.6. Thus the diameter is one. The only graphs with $n > 1$ vertices and diameter 1 are the complete graphs on n vertices. Therefore $\mathcal{T}(C_n) \cong K_n$. □

We saw an example of this in the first tree graph construction in Figure 2 of Section 2.3 when we discovered that $\mathcal{T}(C_4) \cong K_4$.

We say that two graphs are *isoparic* if they have the same number of vertices and edges, but are not isomorphic.

Theorem 3.9. *The tree graph function is not injective.*

Proof. This means us that two nonisomorphic graphs can have the same tree graph. Figure 4 shows an example of two isoparic graphs that have isomorphic tree graphs with 55 vertices and 277 edges. Corollary 3.16 will give a trivial way to break the injectivity of this function, so the importance of this particular counterexample comes from the fact that the two isoparic graphs are 2-connected. □

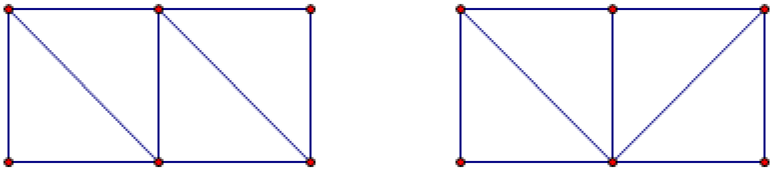


Figure 4: Isoparic graphs with isomorphic tree graphs

The property of being isoparic is not a necessary or sufficient condition for two graphs to have isomorphic tree graphs. For example, both $K_{3,3}$ and $K_{1,1,4}$ have six vertices and nine edges. However, the tree graph of the former has 81 vertices, while the tree graph of the latter has 48. Theorem 3.10 will show another general case where nonisomorphic graphs have the same tree graph.

A graph is *planar* if it can be drawn in the plane with no edge crossings. A *plane* graph is a particular drawing of a planar graph in the plane that contains no edge crossings. There may be many different ways to draw a planar graph as a plane graph. In a plane graph, a *face* is a simply connected region of the plane bounded by at least three edges. For a more complete introduction to these topics, see Chapter 6 in [32].

By convention, G^* denotes the planar dual of G. The dual relationship for planar graphs exchanges the roles of vertices and faces. In particular, G^* is constructed by putting a vertex for every face in a plane graph G, including the infinite outer face. Vertices are connected by an edge each time their corresponding faces share a boundary edge. If G has m edges, n vertices, and f faces, G^* will have m edges, f vertices, and n faces. Figure 5 illustrates the construction of a planar dual. We begin with a plane graph G in (1). In (2) we see the new vertices added for each face of G, and the new edges connecting vertices if their respective faces share a boundary edge. Finally in (3) we see the planar dual G^* redrawn by itself.

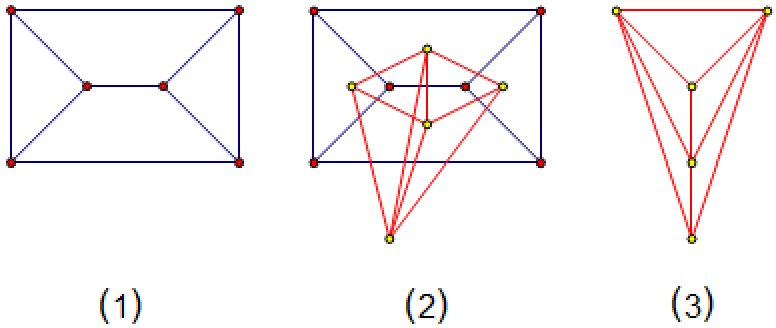


Figure 5: The construction of a planar dual

Theorem 3.10. *If G is 3-connected and planar, then $\mathcal{T}(G) \cong \mathcal{T}(G^*)$.*

Proof. A classic theorem of Whitney [34] says that if G is 3-connected and planar, then G^* is unique up to isomorphism and simple (moreover, also 3-connected and planar). Thus $\mathcal{T}(G^*)$ is well-defined in this case.

A result [18, p. 37] shows that there is a natural bijection between the spanning trees of G and the spanning trees of G^*. Thus $\mathcal{T}(G)$ and $\mathcal{T}(G^*)$ have the same number of vertices. The construction of the bijection [18, p.258] implies that the adjacency relationship between the spanning trees is preserved. That is, $T_i \sim T_j$ in $\mathcal{T}(G)$ if and only if $T_i^* \sim T_j^*$ in $\mathcal{T}(G^*)$. Therefore $\mathcal{T}(G) \cong \mathcal{T}(G^*)$. □

Figure 6 shows an example of this bijection. In (1) we see the starting spanning tree T of a graph G. In (2) we see T in bold along with the rest of G. Next in (3) we see the planar dual G^* drawn in, with the new edges that cross the edges in $E(G) - E(T)$ in bold. Finally in (4) we see the dual tree T^*.

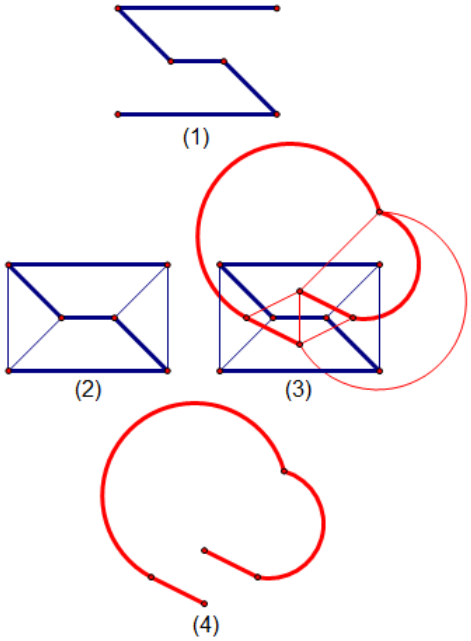


Figure 6: An example of dual spanning tree construction

The cube, with six faces and eight vertices, is the planar dual of the octahedron, with eight faces and six vertices. Figure 7 shows an example of dual spanning trees in these planar dual graphs. The tree graph of these graphs is too large to show here, having 384 vertices and 3768 edges.

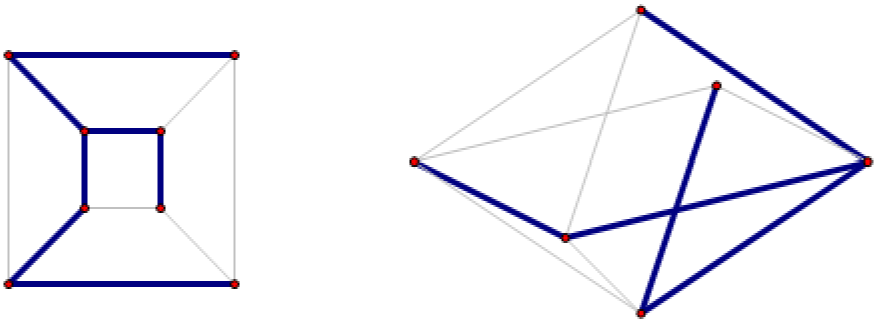


Figure 7: Dual spanning trees (bold edges) in the cube and octahedron graphs

Some examples of 3-connected planar graphs are polyhedral graphs, such as the previously mentioned cube and octahedron, and Halin graphs. To construct a Halin graph, start with a planar drawing of a tree with at least one vertex of degree at least three and no vertices of degree two. Then draw a cycle through all of the leaves in a way that keeps the graph planar. The wheel graph W_n is a special type of Halin graph where the base tree is a star graph, which is a tree with $n-1$ leaves. The following result shows us another situation where nonisomorphic graphs can have isomorphic tree graphs.

Corollary 3.11. *For any non-wheel Halin graph H, we have $H \ncong H^*$ but $\mathcal{T}(H) \cong \mathcal{T}(H^*)$.*

Proof. Euler's formula (see page 241 in [32]) says that for any plane graph, $n-m+f = 2$. Suppose G and G^* are isomorphic. Then $n = f$, which by Euler gives us that $n - m + n = 2$, so $m = 2(n-1)$. Let H be a Halin graph and T_H the spanning tree that induces it. Let l be the number of leaves of T_H. Then $|E(H)| = n - 1 + l$, since the edges of H either come from the set of $n-1$ edges in T_H or from the l-cycle through its leaves. For H^* to be isomorphic to H, we would need $l = n-1$, so that $|E(H)| = n-1+n-1 = 2(n-1)$. But a tree with $n-1$ leaves is a star, so H would be a wheel graph. Thus if $H \cong H^*$, H is a wheel graph. In either case, by Theorem 3.10, $\mathcal{T}(H) \cong \mathcal{T}(H^*)$. $\square$

Figure 8 shows (1) a star graph, (2) the wheel graph W_5, (3) a non-star tree, and (4) the Halin graph built from the tree.

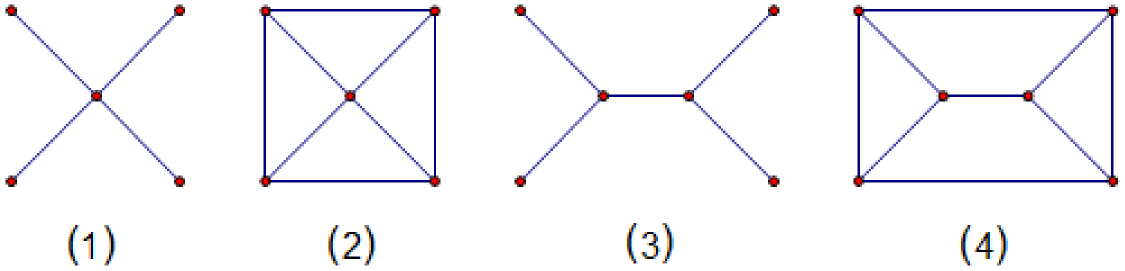


Figure 8: Examples of wheel and Halin graphs

The last three results together tell us that, given a tree graph $\mathcal{T}(G)$, it may not be possible to find a unique graph G that generates it. One step in that direction comes from Lemma 1.2 in [21], which says that if the spanning trees corresponding to any vertex and all of its neighbors in $\mathcal{T}(G)$ are given, the remaining spanning trees can uniquely be assigned to the vertices of $\mathcal{T}(G)$. From this information a unique G can be recovered by taking the union of all of the spanning trees. In general, reconstruction seems to be a challenging problem. That is, given a tree graph $\mathcal{T}(G)$, we want to find a graph H such that $\mathcal{T}(H) \cong \mathcal{T}(G)$. One ambitious desire would be to determine under which circumstances this is possible, and moreover when it is, to know when H is unique.

Let $G \bigodot_{xy} H$ be the graph that identifies the vertex x of G with the vertex y of H. Let the Cartesian product $G \Box H$ be defined as follows. The vertex set is the set

$V(G) \times V(H)$, and two vertices (x, x') and (y, y') are adjacent if and only if either $x = y$ and $x' \sim y'$, or $x' = y'$ and $x \sim y$. The following result shows us that certain tree graphs can be built up as products of smaller tree graphs.

Theorem 3.12. *For any graphs G and H, and vertices $x \in V(G), y \in V(H)$,*

$$\mathcal{T}(G \bigodot_{xy} H) \cong \mathcal{T}(G) \square \mathcal{T}(H).$$

Proof. Let c be the vertex of $G \bigodot_{xy} H$ where G and H are joined at x and y. Then c is a cut vertex of $G \bigodot_{xy} H$. As such, every spanning tree of $G \bigodot H$ has at least two edges incident to c: at least one each from G and H. Thus each tree of $G \bigodot_{xy} H$ can be broken down into a spanning tree of G and a spanning tree of H. Conversely, every pair of spanning trees from G and H, when joined at c, make a spanning tree for $G \bigodot_{xy} H$. Thus $\mathcal{T}(G \bigodot_{xy} H)$ and $\mathcal{T}(G) \square \mathcal{T}(H)$ have the same vertex set.

Let $T_1 \sim T_2$ in $\mathcal{T}(G) \square \mathcal{T}(H)$. This is $(x, y) \sim (x', y')$ for $x, x' \in \mathcal{T}(G)$ and $y, y' \in \mathcal{T}(H)$. By the adjacency rules, WOLOG $x = x'$ and $y \sim y'$ in $\mathcal{T}(H)$. This tells us that y and y' have the edge exchange property. Consider the graphs $x \bigodot y$ and $x' \bigodot y'$. Since $x = x'$, these two graphs differ only by a single edge. Thus they are adjacent in $\mathcal{T}(G \bigodot_{xy} H)$. Using the same argument we can see that adjacent vertices in $\mathcal{T}(G \bigodot_{xy} H)$ will have their corresponding vertices in $\mathcal{T}(G) \square \mathcal{T}(H)$ be adjacent as well. Therefore adjacency is the same in both graphs and they are isomorphic. □

We also have the following corollary, which implies that it does not matter which vertices we choose to identify in the process described above.

Corollary 3.13. *For any $u, x \in V(G)$ and $v, y \in V(H)$,*

$$\mathcal{T}(G \bigodot_{uv} H) \cong \mathcal{T}(G \bigodot_{xy} H).$$

Thus from now on we will simply write $G \bigodot H$ without any reference to the vertices being identified.

Theorem 3.12 was proved independently in [2] (see Lemma 1). The smallest nontrivial example of this is the graph constructed by letting two triangles share a vertex, as shown in Figure 9. In (1) we see two copies of K_3. We then see them joined at a vertex in (2). Finally in (3) we see the tree graph of the middle graph, which is the Cartesian product $\mathcal{T}(K_3)\square\mathcal{T}(K_3) \cong K_3\square K_3$, since $\mathcal{T}(K_3) \cong K_3$ by Lemma 3.8.

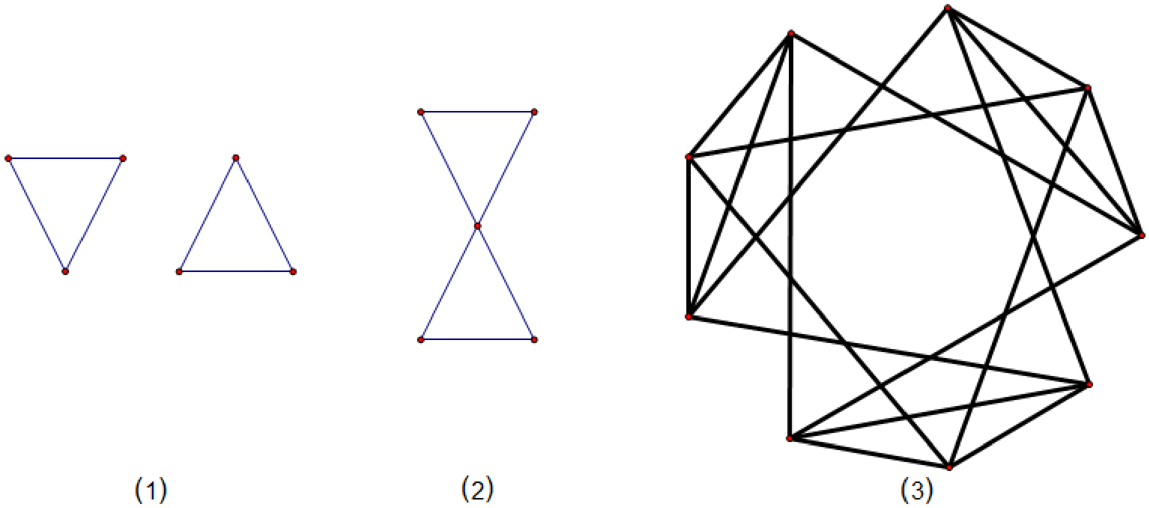


Figure 9: Creating the tree graph of the vertex union of two graphs

Using similar reasoning and a simple induction argument, we get the following two corollaries.

Corollary 3.14. *Fix any $k \geq 1$ and let G be the disjoint union of graphs $H_1, \ldots, H_k$. Then $\mathcal{T}(G) \cong \mathcal{T}(H_1)\square \ldots \square\mathcal{T}(H_k)$.*

Corollary 3.15. *Fix any $k \geq 1$ and graphs $H_1, \ldots, H_k$. Let $G \cong H_1 \bigodot \cdots \bigodot H_k$. Then $\mathcal{T}(G) \cong \mathcal{T}(H_1)\square \ldots \square\mathcal{T}(H_k)$.*

Let $G - e$ denote the graph where the edge e is deleted and $G \cdot e$ denote the graph where the edge has been contracted. When an edge is *contracted*, its endpoints are identified and we follow the convention that the newly formed loop edge is deleted so that simple graphs remain loopless.

For any graph G, let $S = \{e \in E(G) \mid e \text{ is a cut edge}\}$. Define G_{trim} to be the graph remaining after contracting all of the edges in S. If we think of G as being a collection of smaller graphs joined together with the $\bigodot$ operator, we are removing all of the component graphs that are trees. These are the pieces that do not contribute anything new to the tree graph, since they are in every tree. This brings us to our next corollary.

Corollary 3.16. *For any graph G, $\mathcal{T}(G_{trim}) \cong \mathcal{T}(G)$.*

Proof. Suppose $G \cong G_1 \bigodot G_t$, where G_1 is 2-connected and G_t is a tree. By Theorem 3.12, $\mathcal{T}(G) \cong \mathcal{T}(G_1) \Box \mathcal{T}(G_t)$. By Lemma 3.7, $\mathcal{T}(G_t)$ is a single vertex. It is trivial to show that $H \Box K_1 \cong H$ for any graph H. Thus $\mathcal{T}(G) \cong \mathcal{T}(G_1)$. Using Corollary 3.15 we can then conclude that $\mathcal{T}(G_{\text{trim}}) \cong \mathcal{T}(G)$. □

Figure 10 shows a graph G on the left and G_{trim} on the right.

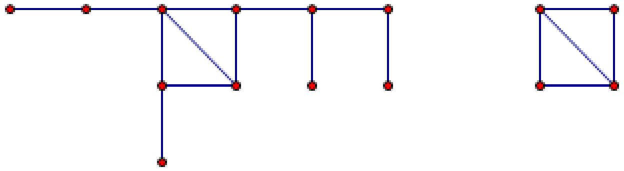


Figure 10: An example of the trimming process

Corollary 3.14 allows us to restrict our attention to the case of connected graphs, as is our working assumption in this chapter. Since connected graphs without cut vertices are 2-connected, Corollary 3.15 allows us to restrict further to consider only the tree graphs of 2-connected graphs. Indeed, the tree graphs of 2-connected graphs are the building blocks of all tree graphs, just as primes are the building blocks of the natural numbers. Accordingly, every time a graph is mentioned as being the input for the tree graph function in this paper, we will henceforth assume that it is 2-connected unless otherwise noted.

3.2 Parameters of Tree Graphs

In this section we collect results concerning parameters of tree graphs $\mathcal{T}(G)$, especially in their relation to the parameters of G.

Recall that $\kappa(G)$ is the *vertex connectivity* of a graph: for a non-complete graph, the minimum number of vertices needed to be removed in order to disconnect it. Likewise, $\kappa'(G)$ is the edge connectivity. For more information on these parameters, see Chapter 4 in [32]. The smallest degree of a graph is $\delta(G)$. A classic result is that the following chain of inequalities holds [33]:

$$\kappa(G) \leq \kappa'(G) \leq \delta(G).$$

One way to get a feel for these inequalities is that deleting all of the neighbors of a vertex or all of the edges incident to a vertex will separate that vertex from the rest of the graph. Additionally, deleting an edge from a graph does not delete its incident vertices in the way that deleting a vertex also removes its incident edges. The following theorem of Liu highlights a special property of tree graphs.

Theorem 3.17. *[16, Corollary 2.8] For all graphs G,*

$$\kappa(\mathcal{T}(G)) = \kappa'(\mathcal{T}(G)) = \delta(\mathcal{T}(G)).$$

This theorem tells us that tree graphs are maximally connected. This can be an important property to network designers, as local disruptions should not jeopardize the entire structure.

A *coloring* of a graph is an assignment of numbers to its vertices. A *proper coloring* is a coloring where adjacent vertices get different numbers. The *chromatic number* of a graph, $\chi(G)$, is the minimum number of colors in a proper coloring of G. See Chapter 5 in [32] for more information on this parameter.

The following result gives us an upper bound on the chromatic number of a tree graph based on the number of edges of the underlying graph.

Theorem 3.18. *[26, Theorem 1] For all graphs G,*

$$\chi(\mathcal{T}(G)) \leq |E(G)|.$$

As an example, $K_4 - e$ has five edges, so the tree graph $\mathcal{T}(K_4 - e)$ needs at most five colors for a proper coloring. It contains a K_4 subgraph, so requires at least four colors. Figure 11 demonstrates that this is sufficient.

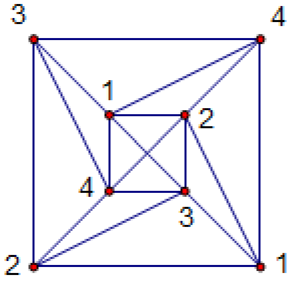


Figure 11: A proper four-coloring of $\mathcal{T}(K_4 - e)$

The *girth* of a graph, denoted girth(G), is the length of its shortest cycle. The *circumference* of a graph, denoted circ(G), is the length of it longest cycle. This next result gives us a lower bound on the smallest degree (δ) and an upper bound on the largest degree (Δ) of a tree graph $\mathcal{T}(G)$ based on cycle and edge information from the underlying graph G.

Theorem 3.19. *[17, Theorem 1] For a graph G with n vertices and m edges, let $v = m - n + 1$. Then*

$$v \cdot (\mathrm{girth}(G) - 1) \leq \delta(\mathcal{T}(G)) \leq \Delta(\mathcal{T}(G)) \leq v \cdot (\mathrm{circ}(G) - 1).$$

In our $K_4 - e$ example, we have $m = 5$ and $n = 4$, so $v = 2$. The length of the shortest cycle is three while the length of the longest cycle is four. Putting these together gives us that $4 \leq \delta(\mathcal{T}(K_4 - e))$ and $\Delta(\mathcal{T}(K_4 - e)) \leq 6$. Compare these to the actual values of $\delta(\mathcal{T}(K_4 - e)) = 4$ and $\Delta(\mathcal{T}(K_4 - e)) = 5$.

The *chromatic index* of a graph, $\chi'(G)$, is the edge coloring version of the chromatic number. We need a minimum of $\Delta(G)$ distinct colors to properly color the edges of G. A famous theorem by Vizing [29] says that at most one more color is necessary. Together with Theorem 3.19, this gives us the following corollary.

Corollary 3.20. *For a graph G with n vertices and m edges, let $v = m-n+1$. Then*

$$\chi'(\mathcal{T}(G)) \leq (\mathrm{circ}(G) - 1) \cdot v + 1.$$

If we know something about the structure of the trees, we can give an exact value to the minimum degree of the tree graph. A *unicycle* is a connected graph with exactly one cycle. All unicycles in this paper will be assumed to be spanning.

Theorem 3.21. *For a graph G with n vertices and m edges, let $v = m - n + 1$.*

(i) There exists a tree of G such that every unicycle that contains it has cycle length equal to girth(G) *if and only if $\delta(\mathcal{T}(G)) = v \cdot (\mathrm{girth}(G) - 1)$.*

(ii) There exists a tree of G such that every unicycle that contains it has cycle length equal to circ(G) *if and only if $\Delta(\mathcal{T}(G)) = v \cdot (\mathrm{circ}(G) - 1)$.*

Proof. (i) Suppose first that such a tree existed. Then by Theorem 4.22 it will achieve degree $v \cdot (\text{girth}(G) - 1)$, which by Theorem 3.19 is as small as possible. Thus $\delta(\mathcal{T}(G)) = v \cdot (\text{girth}(G) - 1)$.

Now suppose $\delta(\mathcal{T}(G)) = v \cdot (\text{girth}(G) - 1)$. There is some vertex x in $\mathcal{T}(G)$ with that degree. Again by Theorem 4.22, x is part of v cliques. Each $c-$clique contributes $c - 1$ edges to the degree of x. Let $s_1, s_2, \ldots, s_v$ be the number of edges added to the degree of x for each of the v cliques that contain x. Since each clique in $\mathcal{T}(G)$ comes from a cycle in G, the smallest a clique can be is the size of the smallest cycle, which is girth(G). Thus $s_i \geq \text{girth}(G) - 1$. We have that $s_1 + s_2 + \cdots + s_v = v \cdot (\text{girth}(G) - 1)$, so the average value of the s_i is $(\text{girth}(G) - 1)$. Since they are all positive integers bounded by their average, they must equal that average. Thus x is only part of cliques of size girth(G), so all unicycles containing x have cycle length equal to girth(G).

(ii) The proof is similar to that of (i). □

The graph $K_4 - e$ is an example that has such a tree for the lower bound. The value girth$(K_4 - e) = 3$ and $v = 5 - 4 + 1 = 2$. Any tree that contains the diagonal edge cannot be part of a unicycle with cycle size 4. The only other possible cycle size is 3, so all unicycles containing that tree have cycle size 3. Thus in $\mathcal{T}(K_4 - e)$ we see four vertices of minimum degree $2 \cdot (3 - 1) = 4$. The house graph, shown in Figure 12, has girth 3 and is a nonexample. The tree graph of the house graph has minimum degree 5, which is not a multiple of 2, which is the v value for the house. Therefore no tree of the house can contain only unicycles with cycle size 3.

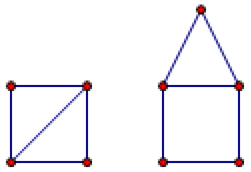


Figure 12: $K_4 - e$ and the house graph

It is easier to find graphs that meet the lower bound than the upper bound. The bound on Δ is sharp for some families, such as the $\theta_{a,2,2}$ and $P_{n,k}$ graphs that are defined in Section 4.8.

The *clique number* of a graph, $\omega(G)$, is the size of its maximum clique. We can place a lower bound on the clique number of a tree graph based on the circumference of the base graph.

Theorem 3.22. *For any graph G,*

$$\omega(\mathcal{T}(G)) \geq \operatorname{circ}(G).$$

Proof. Let $s = \operatorname{circ}(G)$. The cycle C_s is then a subgraph of G. By Corollary 4.19, $\mathcal{T}(C_s)$ will be a subgraph of $\mathcal{T}(G)$. By Lemma 3.8, $\mathcal{T}(C_s) \cong K_s$. Thus $\mathcal{T}(G)$ will contain a K_s subgraph, and so $\omega(\mathcal{T}(G))$ is at least $\operatorname{circ}(G)$. □

For most of the tree graphs studied, this bound is tight. However, for some families of graphs the gap between the two values can get arbitrarily large. For example, the $K_{n,2}$ graphs have a circumference of 4. Call the vertices in the two-cell x and y. Consider the set of trees where x is adjacent to all of the vertices in the n-cell and y is adjacent to just one of them. There are n such trees and they all differ by just one edge, so they will all be adjacent in $\mathcal{T}(K_{n,2})$. Thus $\omega(\mathcal{T}(K_{n,2})) \geq n$, which can be taken as far away from 4 as we want.

Each vertex in a clique needs a different color in a proper coloring. This gives us the easy bound $\chi(G) \geq \omega(G)$. From this fact and Theorem 3.22 we get the following corollary.

Corollary 3.23. *For any graph G,*

$$\chi(\mathcal{T}(G)) \geq \text{circ}(G).$$

For almost all of the tree graphs studied, the clique number and the chromatic number were the same. One time where these differed was when the base graph was K_4. We have that $\omega(\mathcal{T}(K_4)) = 4 < 5 = \chi(\mathcal{T}(K_4))$. Figuring out exactly when these values coincide would be valuable.

A *regular* graph has vertices all of the same degree. The *independence number* of a graph is $\alpha(G)$, defined to be the maximum number of vertices in a graph that induce a subgraph with no edges. Let G have n vertices and m edges. Let $\mu(G)$ be the number of spanning unicycles in G, and let $v = m - n + 1$. Let $P_{n,k}$ be the graph where two vertices are joined by n openly-disjoint paths of length k, $n, k > 1$. Paths are openly-disjoint if and only if they share only their endpoints.

In the proof for Theorem 3.22 we saw that each unicycle with cycle size c in G gives rise to a $c-$clique in $\mathcal{T}(G)$. Later on, Theorem 4.22 will show us that $\mathcal{T}(G)$ can be decomposed into $\mu(G)$ cliques such that each vertex is part of exactly v cliques. At most one vertex from each of those cliques can be chosen to be part of an independent set. Thus, choosing a vertex prevents one from choosing any other vertices from the v cliques it is part of. So the number of cliques in a decomposition divided by the number of cliques per vertex should give us an upper bound on the number of independent vertices we can choose. That brings us to our next result.

Theorem 3.24. *For a graph G with n vertices and m edges, let $v = m - n + 1$. Then*

$$\alpha(\mathcal{T}(G)) \leq \lfloor \mu(G)/v \rfloor.$$

Proof. Let $S = \{s_1, s_2, \ldots, s_\alpha\}$ be the vertices in a maximum independent set in $\mathcal{T}(G)$. We count the ordered pairs of the form (s_i, c_i), where c_i is a clique in the clique decomposition from Theorem 4.22 that contains s_i. If we choose the vertices first, we have α choices. Each vertex is in v cliques, so we have that many choices for c_i. Thus we have $\alpha \cdot v$ ordered pairs. Suppose that $\alpha \cdot v > \mu(G)$. Since we only have $\mu(G)$ different cliques, by the pigeonhole principle at least one of them is repeated in our set of ordered pairs. That implies that we have chosen at least two vertices from the same clique, which contradicts the fact that the vertices are chosen from an independent set. Thus we have that $\alpha \cdot v \leq \mu(G)$. Since the independence number is an integer, we get the final bound of $\alpha(\mathcal{T}(G)) \leq \lfloor \mu(G)/v \rfloor$. □

For example, if the base graph is $K_{1,1,3}$, we have $\mu = 18$ and $v = 3$, while $\lfloor \frac{18}{3} \rfloor = 6 \geq 5 = \alpha$. For another, let $G = P_{3,2}$, which is 6-regular. Then we have $\mu = 6$ and $v = 2$, which gives the correct value of $\alpha = 3$. If $\mathcal{T}(G)$ is regular, this bound seems to be tight.

4 Properties of Tree Graphs

This section will contain results concerning structural properties that all tree graphs share. Some concern graph-theoretic properties, while others are more algebraic in nature. Together they help demonstrate how truly remarkable the family of tree graphs is. Throughout Chapter 4, we assume graphs are connected unless otherwise stated.

4.1 Paths and Cycles

A graph is *hamiltonian* if it has a cycle that contains all of its vertices. When we investigate this property for tree graphs, we are really looking into whether or not it is possible to cycle through all of the spanning trees of a graph by just swapping one edge at a time. The following result, from the first paper on tree graphs, says this is always possible.

Theorem 4.1. *[4, Theorem L] For any graph G, $\mathcal{T}(G)$ is hamiltonian.*

Figure 13 shows one possible hamiltonian cycle through $\mathcal{T}(K_4 - e)$. The hamiltonicity of tree graphs might be desirable to circuit or network designers who need to test the performance of every possible spanning tree of their system. Some authors, such as that of [11], even came up with constructive algorithms to generate and cycle through all of the spanning trees of a graph in this way.

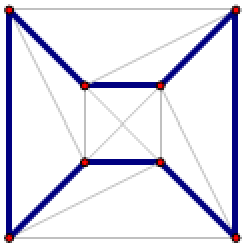


Figure 13: A hamiltonian cycle through $\mathcal{T}(K_4 - e)$

A graph is *uniformly hamiltonian* if for each of its edges e, there exists a hamiltonian cycle that uses e and one that avoids e.

Theorem 4.2. *[10, Corollary 2] For any graph G, the tree graph $\mathcal{T}(G)$ is uniformly hamiltonian.*

So while $K_4 - e$ is hamiltonian, it is not uniformly hamiltonian, as no hamiltonian cycle can use the diagonal edge. This result gives us that $\mathcal{T}(K_4 - e)$, however, is uniformly hamiltonian. Figure 14 shows an example of this for one edge, e'.

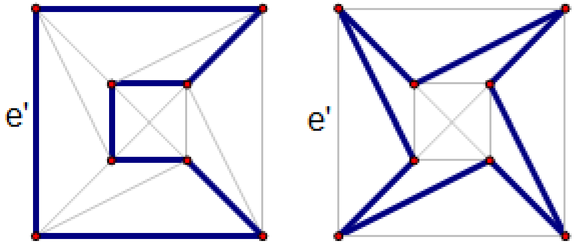


Figure 14: Hamiltonian cycles using and avoiding edge e' in $\mathcal{T}(K_4 - e)$

A graph is *edge-pancyclic* if each of its edges is used in a cycle of every possible size from its girth to its circumference. *Hamiltonian-connected* means that there is a hamiltonian path between every two vertices in the graph. *Path-full* implies that if there exists paths of length m and n between two vertices, then for all $m < k < n$ there exists a path of length k between them as well.

Theorem 4.3. *[1, Theorems 3.1-3] $\mathcal{T}(G)$ is edge-pancyclic, hamiltonian-connected, and path-full for any graph G.*

These facts are illustrated for $\mathcal{T}(K_4 - e)$ in the next few figures.

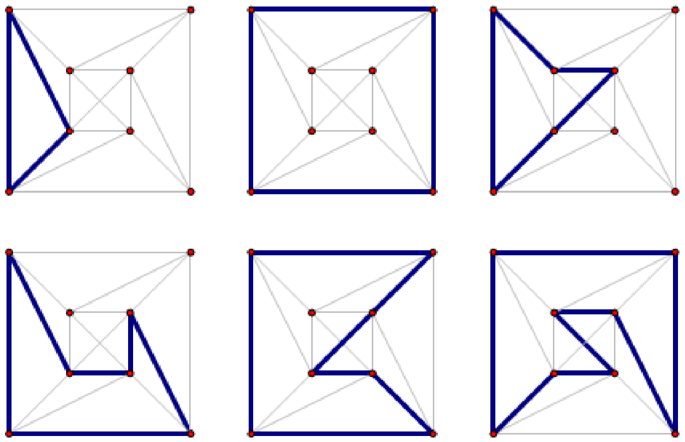


Figure 15: Cycles of various lengths in $\mathcal{T}(K_4 - e)$

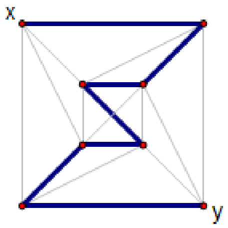


Figure 16: A hamiltonian path between vertices x and y in $\mathcal{T}(K_4 - e)$

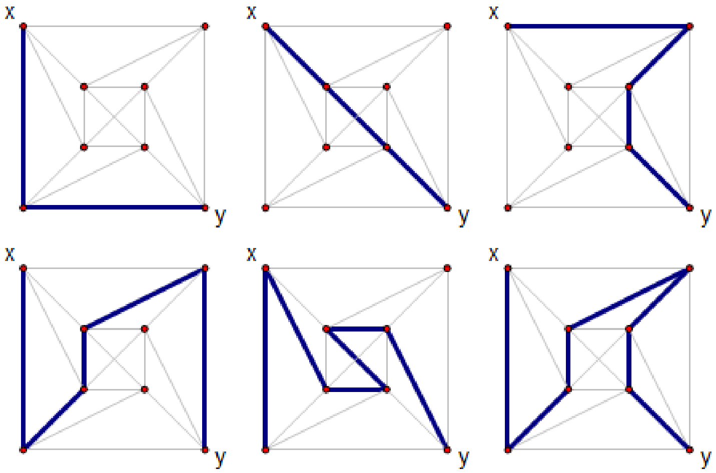


Figure 17: Various xy-paths in $\mathcal{T}(K_4 - e)$

Together these results show us that tree graphs are very structured and have plenty of edges with which to move through the graph.

4.2 Centers

A subgraph $H = (V', E')$ of $G = (V, E)$ is an *induced subgraph* if for every two vertices x and y in V', $x \sim y$ in H if and only if $x \sim y$ in G. In general, the *complement* G^c has the same vertices as G but with $x \sim y$ in G^c if and only if $x \nsim y$ in G. For our purposes, we need a second type of complement. Specifically, if $H \leq G$, then we define $\overline{H} = (V(H), E(G) - E(H))$. In particular, the complement of a spanning tree T of G will be understood to be the complement of T in G.

Recall that the *center* of a graph, denoted $C(G)$, is the subgraph induced on the set of vertices of minimum eccentricity. In some sense, these are the vertices that are closest to all other vertices. If $G \cong C(G)$, we say that G is *self-centered.* An equivalent definition is to say that $\operatorname{rad}(G) = \operatorname{diam}(G)$. Some tree graphs are self-centered while others have a proper center. The distinction between the two cases is explored in the next results.

Lemma 4.4. *Let T be a spanning tree of G. If $\overline{T}$ is acyclic, then the eccentricity of the vertex in $\mathcal{T}(G)$ corresponding to T is given by*

$$\operatorname{ecc}(T) = m - n + 1.$$

Proof. Every tree has $n - 1$ edges, so $\overline{T}$ has $m - n + 1$ edges. By Lemma 2.6, $\overline{T}$ is contained in a spanning tree of G. Lemma 3.2 tells us this tree will be at distance $m - n + 1$ from T, which by Lemma 3.3 is the maximum possible. □

Corollary 4.5. *If all trees of G have acyclic complements, $\mathcal{T}(G)$ is self-centered.*

Proof. By Lemma 4.4 all of the trees will correspond to vertices in $\mathcal{T}(G)$ with the same eccentricity, which implies $\operatorname{rad}(\mathcal{T}(G)) = \operatorname{diam}(\mathcal{T}(G))$ and that $\mathcal{T}(G)$ is self-centered. □

Since all cycles have at least three edges, if $v = m - n + 1 = 2$ no tree complement can contain a cycle. Thus all graphs with $v = 2$ have self-centered tree graphs. This family includes the $P_{n,k}$ graphs and the $\theta_{a,b,c}$ graphs, described later in this chapter.

A graph is *bipartite* if its vertices can be partitioned into two sets such that every edge in the graph has one end in each set. Bipartite graphs have no odd cycles. Thus if G is bipartite and $v = 3$, $\mathcal{T}(G)$ will be self-centered. In general, we get the following corollary.

Corollary 4.6. *Let G be any graph, and let $v = m - n + 1$. If $v < \operatorname{girth}(G)$, then $\mathcal{T}(G)$ will be self-centered.*

The converse of this corollary is false. Consider the graph $G = K_{4,1,1}$. Since it has only two vertices of degree higher than two, every cycle in it contains at least one vertex of degree two. Suppose the complement of a spanning tree T of G contained a cycle. The complement $\overline{T}$ would then contain all of the edges incident with at least one of the degree two vertices. Thus in T that vertex would have no edges incident to it, contradicting the fact that T is a spanning tree. Therefore all trees of G have acyclic complements, and so by Corollary 4.5 the graph $\mathcal{T}(G)$ is self-centered. However, we have that $\operatorname{girth}(G) = 3$ and $v(G) = 4$.

As the inverse to Lemma 4.4, we have the following.

Lemma 4.7. *Let T be a spanning tree of G. If $\overline{T}$ contains a cycle, then the eccentricity of the vertex in $\mathcal{T}(G)$ corresponding to T satisfies*

$$\operatorname{ecc}(T) < m - n + 1.$$

Proof. No tree in $\mathcal{T}(G)$ contains $\overline{T}$, since it contains a cycle. This implies $\mathrm{d}(T', T) < m - n + 1$ for all $T' \in \mathcal{T}(G)$, since no tree can be $m - n + 1$ away from T by using all of the edges in its complement. Thus $\mathrm{ecc}(T) < m - n + 1$. □

Combining the last few results, we can now describe the trees in the center of a non-self-centered graph.

Theorem 4.8. *If $\mathcal{T}(G)$ is not self-centered and $T \in C(\mathcal{T}(G))$, then $\overline{T}$ contains a cycle.*

Proof. Since the center is proper, not all trees in $\mathcal{T}(G)$ have the same eccentricity. Thus by Corollary 4.5 at least one tree T' has a complement that contains a cycle. No tree with an acyclic complement can be in the center, since by Lemmas 4.4 and 4.7 its eccentricity would be strictly greater than that of T'. Thus all trees in the center have a complement that contain a cycle. □

Figure 18 shows an example of the previous results on tree graph centers. In (1) we see our graph G. It has 6 vertices and 8 edges, and so $v = 3$. In (2) and (3) we see trees T_1 and T_2 in bold. Note that (4) shows $\overline{T_1}$, which contains a triangle. On the other hand, (5) shows $\overline{T_2}$, which is acyclic. Thus trees can be at most distance two from T_1 and distance three from T_2 in $\mathcal{T}(G)$. Since there is only one triangle in G and that is the only cycle that can be formed with $v = 3$ edges, T_1 is the only vertex in the center of $\mathcal{T}(G)$.

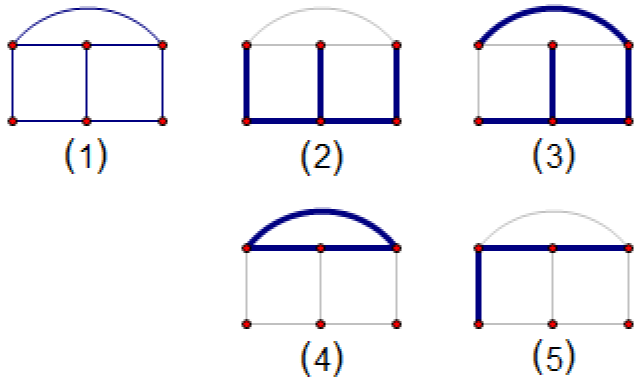


Figure 18: Trees and their complements determine the center of $\mathcal{T}(G)$

The converse of Theorem 4.8 is also not true. Consider $G = K_5$. The center of $\mathcal{T}(G)$ consists of the five spanning trees that are star graphs. Let T be any path through G. The complement $\overline{T}$ contains a triangle between the two endpoints of the path and the middle vertex of the path. Therefore $\overline{T}$ contains a cycle, but T is not in the center of $\mathcal{T}(G)$.

The next result, from the literature, relates the diameter of self-centered graphs with the diameter of their line graphs. A graph is *d-self-centered* if it is self-centered with diameter d. We define $\text{Grid}_{a,b}$ to be the graph $P_a \square P_b$.

Theorem 4.9. *[28, Theorem 2.2] Let G and $L(G)$ be d- and f-self-centered graphs, respectively. Then $f \in \{d-1, d, d+1\}$.*

Thus if a graph and its line graph are both self-centered, their diameters can differ by at most one. A noteworthy passage in [28] poses the following:

> If $G = K_2$ (1-self-centered) then $L(G) = K_1$ is 0-self-centered. On the other hand, examples of G and $L(G)$ which are both d-self-centered can be easily found. Hence we have the following interesting problem: *Determine the remaining (if any) d-self-centered graphs whose line graphs are $(d-1)$- or $(d+1)$-self-centered.*

Several tree graphs are examples of the latter type. The tree graphs of $K_{3,2}, K_{4,2}$, and $\text{Grid}_{3,2}$ all have line graphs that are also self-centered whose diameter has increased by one. We conjecture that there is an infinite family of tree graphs with this property, but it has not been fully investigated yet.

4.3 Local Properties

An ordered pair $(E, \mathcal{B})$ is a *matroid* if E is a set of *elements* and $\mathcal{B} \subseteq \mathcal{P}(E)$ is a set of subsets of E called *bases* satisfying: (i) If $B \in \mathcal{B}$ then $B \neq \emptyset$ and (ii) If $A, B \in \mathcal{B}$

then for all $a \in A - B$ there exists $b \in B - A$ such that $(A - \{a\}) \cup \{b\} \in B$. It turns out that the spanning trees of a graph can be used to form a matroid. If we let E be the set of edges of a graph G, and $\mathcal{B}$ be the set of spanning trees of G, then $M_G = (E, \mathcal{B})$ is a matroid.

Given any matroid M, the *matroid basis graph* is the graph whose vertices are the bases of the matroid and where two vertices are adjacent if and only if their bases differ by a single element. Notice that this sounds remarkably like our definition of a tree graph. In fact, when the collection of spanning trees of a graph is viewed as a matroid as above, tree graphs can be seen as a special case of matroid basis graphs. Thus results that hold for matroid basis graphs also apply to tree graphs.

Tree graphs constitute a proper subclass of the class of all matroid basis graphs, however. For example, let $M = \{\{1, 2, 3, 4\}, \{12, 13, 14, 23, 24, 34\}\}$. The matroid basis graph for M is the octahedron (see Figure 7). Since the bases all have two elements, if they were the edges of a tree, the base graph would need to have exactly three vertices. But our edge set has four elements. Indeed, there are no simple graphs with three vertices and four edges. Thus, this matroid basis graph is not a tree graph.

Maurer published influential papers [20, 21] that described and characterized matroid basis graphs. Many local and global properties were investigated. One concept in these papers is that of the common neighbor graph, defined as follows. Let x and y be vertices of G that are at distance two. Let $N = \{v \in V \mid v \sim x, v \sim y\}$ be the set of common neighbors of x and y. The *common neighbor graph* of x and y is the subgraph of G induced on the set of vertices $N \cup \{x, y\}$. A *square* is the graph C_4 and the *pyramid* is the wheel graph W_5.

Theorem 4.10. *All common neighbor graphs of tree graphs are squares or pyramids.*

Proof. A matroid is *graphic* if its bases are the edge sets of the spanning forests of some graph. Thus the set of spanning trees of a connected graph together with the set of edges of the graph is a graphic matroid. A remark in [21, p. 125] states that every graphic matroid is binary (another designation for matroids, whose definition is not important for our purposes), which, in turn, is equivalent to the basis graph having no induced octahedra (see Theorem 4.1 in [21]). Thus $\mathcal{T}(G)$, as the basis graph for the set of spanning trees of G, contains no induced octahedra. But Lemma 1.4 in [20] says that the common neighbor graph of a basis graph is either a square, pyramid, or an octahedron. Since we just learned that octahedra are ruled out for tree graphs, all common neighbor graphs of tree graphs are either squares or pyramids. Additionally, Corollary 4.3 in [21] states that if G is not a Cartesian product of complete graphs, then $\mathcal{T}(G)$ contains at least one common neighbor graph that is a pyramid. □

Figure 19 shows square and pyramid common neighbor graphs in $\mathcal{T}(K_4 - e)$.

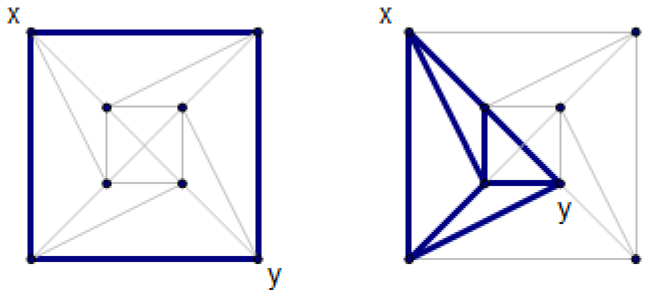


Figure 19: Common neighbor graphs in $\mathcal{T}(K_4 - e)$

4.4 Homomorphisms

A *homomorphism* from graph G to H is a function $\phi : V(G) \to V(H)$ such that $x \sim y \Rightarrow \phi(x) \sim \phi(y)$. If there is a homomorphism from G to H, we will write $G \to H$. Homomorphisms preserve some of the structure of the original graph. In this section we make a short investigation of how the tree graph function interacts with graph homomorphisms.

A natural first question about homomorphisms is to determine when a homomorphism can exist between graphs. One necessary condition for $G \to H$ is that $\chi(G) \leq \chi(H)$. In any proper coloring, distinctly colored vertices cannot be adjacent, so H cannot require fewer colors than G needs. From this observation we get the following result.

Theorem 4.11. *The assumption that $G \to H$ does not imply $\mathcal{T}(G) \to \mathcal{T}(H)$.*

Proof. By way of a counterexample, notice that there is a homomorphism from $K_4 - e$ to K_3 that maps the two vertices of degree 2 to a single vertex of K_3 and the two vertices of degree 3 to the two other vertices in K_3. However, $\chi(\mathcal{T}(K_4 - e)) = 4$, while $\chi(\mathcal{T}(K_3)) = 3$, so there can be no homomorphism from $\mathcal{T}(K_4 - e)$ to $\mathcal{T}(K_3)$. Therefore $G \to H$ does not imply $\mathcal{T}(G) \to \mathcal{T}(H)$. □

4.5 Automorphism Groups

We denote by $\mathrm{Aut}(G)$ the the group of automorphisms of G, which is the group of permutations of $V(G)$ that respect adjacency in G. The *glory* $g(G)$ of a graph G is the size of its automorphism group, so that $g(G) = |\mathrm{Aut}(G)|$. (This definition gives us a way to quantify how glorious a graph is!) Automorphism groups can range in size anywhere from the full symmetric group S_n (of order $n!$) for complete graphs, all the way down to the trivial group (of order 1) for some graphs. Such inglorious graphs are called *asymmetric*. It is known that, asymptotically, almost all finite graphs are asymmetric (see Corollary 2.3.3 in [9]).

The data gathered for this research indicate that tree graphs tend to be very glorious. This high level of symmetry can arise even if the base graph is asymmetric. For example, the smallest 2-connected asymmetric graph can be visualized (see Figure 20) by identifying an edge of two squares and adding a diagonal edge through one of the squares. We will refer to this graph as Asym6. Its tree graph has 29 vertices and 122 edges, and has an automorphism group of order 12. In fact, the automorphism group is isomorphic to D_{12}, the symmetries of a regular hexagon.

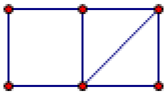


Figure 20: An asymmetric graph G with $|\mathrm{Aut}(\mathcal{T}(G))| = 12$

Brendan McCay's well-known computer program *nauty* [22] was used to find the glories of the tree graphs constructed during this research. In some smaller cases, the automorphism group itself could be uncovered. The number of groups discerned so far is perhaps not large enough to support any serious conjecture, but all of the groups found have either been dihedral groups or (products of) symmetric groups. This only applies to graphs that are 2-connected but not 3-connected, a distinction that will be explained in Conjecture 4.33.

The two major results in this section relate the automorphism group of a tree graph $\mathcal{T}(G)$ to the automorphism group of G. In Theorem 4.15 we see that the latter is always isomorphic to a subgroup of the former. In Theorem 4.16 we learn that in most cases, if G is sufficiently connected then the two groups are the same. We first prove several lemmas. We remind the reader of Menger's theorem [23] on connectivity which says that if G has connectivity κ, then any two vertices x, y will be connected by at least κ openly-disjoint xy-paths (xy-paths are openly-disjoint if and only if they share only their endpoints x and y).

Lemma 4.12. *Let G be a 2-connected graph and let $e, e' \in E(G)$ be distinct. There is a cycle in G containing both e and e'.*

Proof. By the fact that G is 2-connected, we have that $\kappa \geq 2$. Suppose that e and e' share a vertex, that is, $e = \{x, y\}$ and $e' = \{x, z\}$. By Menger's Theorem there exists a path from y to z openly-disjoint from the path yxz. The union of these two paths then creates a cycle containing e and e'.

Suppose now that $e = \{w, x\}$ and $e' = \{y, z\}$ share no vertices. Since the graph is connected, there exists a path p from w to y. This path falls into one of several cases.

- the path p contains neither x nor z: by Menger's theorem there exists a path q from x to z openly-disjoint from p.
- the path p contains x but not z: by Menger's theorem there exists a path q from w to z openly-disjoint from p.
- the path p contains z but not x: by Menger's theorem there exists a path q from x to y openly-disjoint from p.
- the path p contains x and z: by Menger's theorem there exists a path q from w to y openly-disjoint from p.

In each case, the union of p, q, e, and e' produces a cycle containing the two desired edges. □

Figure 21 illustrates each of the four cases. The bold edges represent the path p and the dashed edges represent the path q.

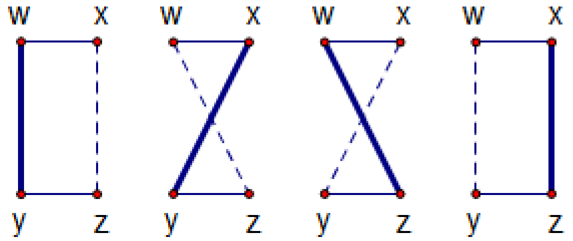


Figure 21: Building a cycle that contains e and e'

Lemma 4.13. *Let G be a 2-connected graph and let $e, e' \in E(G)$ be distinct. There exists a spanning tree of G which includes e and avoids e'.*

Proof. By Lemma 4.12, there exists a cycle containing both e and e'. Start building a tree by first adding in all of the edges in that cycle except for e'. That collection of edges forms an acyclic subgraph of G. By Lemma 2.6, there is a tree that contains that subgraph. Thus we have a tree that contains e and does not contain e'. □

Let $\sigma \in \text{Aut}(G)$. Then σ is a permutation of $V(G)$ such that $\sigma(x) \sim \sigma(y)$ if and only if $x \sim y$. But σ also induces a permutation $\hat{\sigma}$ of edges of G. We define $\hat{\sigma} : \mathcal{P}(E(G)) \to \mathcal{P}(E(G))$ to be the map satisfying

$$\hat{\sigma}(S) = \{\{\sigma(x), \sigma(y)\} \mid \{x, y\} \in S\}.$$

We will view T, a spanning tree of G, both as a vertex of $\mathcal{T}(G)$ and as a set of edges of G, depending on our need. By the adjacency restriction of σ on $V(G)$, we know that $\hat{\sigma}(T) \cong T$ as a spanning tree. The function σ also induces an automorphism $\phi_\sigma \in \text{Aut}(\mathcal{T}(G))$ by $\phi_\sigma(T) = \hat{\sigma}(T)$.

There are two types of edges: those that are not contained in cycles (so-called *cut edges*) and those that are (so-called *cycle edges*). Since automorphisms preserve the structure of the graph, the edge orbits under $\text{Aut}(G)$ are partitioned by these types. That is, cycle edges get sent only to cycle edges and cut edges get sent only to cut edges under the action of $\text{Aut}(G)$.

Lemma 4.14. *Let $\sigma \in \text{Aut}(G)$ and ϕ_σ be defined as above. If $\phi_\sigma(T) = T$ for all $T \in V(\mathcal{T}(G))$, then $\hat{\sigma}(e) = e$ for all cycle edges $e \in E(G)$.*

Proof. This theorem says that if an induced automorphism acts like the identity on $\mathcal{T}(G)$, then its base automorphism fixes cycle edges in G. We argue by contrapositive.

Suppose that $\hat{\sigma}$ does not fix all cycle edges in G. Then there exist distinct cycle edges e and e' such that $\hat{\sigma}(e) = e'$. By Lemma 4.13, there exists a spanning tree T

of G that contains e and avoids e'. Then $\phi_\sigma(T) \neq T$, since $\hat{\sigma}(T)$ contains e' while T does not. Therefore the statement is true by contrapositive. □

Theorem 4.15. *For any 2-connected graph G, there exists a subgroup $H \leq \mathrm{Aut}(\mathcal{T}(G))$ such that $\mathrm{Aut}(G) \cong H$.*

Proof. Using the same notation as above, define $\Psi : \mathrm{Aut}(G) \to \mathrm{Aut}(\mathcal{T}(G))$ by $\Psi(\sigma) = \phi_\sigma$ for any $\sigma \in \mathrm{Aut}(G)$. We claim that Ψ is a group homomorphism. To see this, let $\sigma_1, \sigma_2 \in \mathrm{Aut}(G)$ and $T \in V(\mathcal{T}(G))$. Then

$$\begin{aligned}\Psi(\sigma_2 \circ \sigma_1)(T) &= \phi_{\sigma_2 \circ \sigma_1}(T)\\ &= \hat{\sigma}_2(\hat{\sigma}_1(T))\\ &= \phi_{\sigma_2}(\phi_{\sigma_1}(T))\\ &= \Psi(\sigma_2) \circ \Psi(\sigma_1)(T).\end{aligned}$$

Thus Ψ is a homomorphism. We will now show that Ψ is an injection. Let i be the appropriate identity automorphism. If $\Psi(\sigma) = \Psi(i)$ then for any $T \in V(\mathcal{T}(G))$,

$$\hat{\sigma}(T) = \phi_\sigma(T) = \Psi(\sigma)(T) = \Psi(i)(T) = \phi_i(T) = \hat{i}(T).$$

This implies that $\hat{\sigma}$ equals $\hat{i}$ when they are acting on $V(\mathcal{T}(G))$. But we want to show that σ equals i as elements of $\mathrm{Aut}(G)$, ie. when they are acting on $V(G)$.

We are assuming that $\phi_\sigma(T) = T$ for all $T \in V(\mathcal{T}(G))$. Since G is 2-connected, every edge is a cycle edge. By Lemma 4.14, this means that $\hat{\sigma}(e) = e$ for all $e \in E(G)$. Thus $\hat{\sigma}$ fixes all of the edges of G.

By way of contradiction, then, suppose σ does not fix all of the vertices in G, that is, $\sigma(x) = y$, for some $x \neq y$. If $\{w, x\}$ is an edge that contains x, then either

$\hat{\sigma}(\{w,x\}) = \{\sigma(w), y\}$ is a different edge, which is a contradiction, or else $w = y$ and $\sigma(y) = x$. Since G is 2-connected, it contains more than just the one edge $\{x,y\}$ with endpoint x. Indeed, let $\{x,z\}$ denote another edge in G. The transposition of x and y forces z to move as well, which sends $\{x,z\}$ to a different edge. This is a contradiction.

Thus we have that σ fixes all of the vertices of G, meaning $\sigma = i$. This implies that Ψ is an injective group homomorphism from $\mathrm{Aut}(G)$ into $\mathrm{Aut}(\mathcal{T}(G))$, which lets us conclude that the image of $\mathrm{Aut}(G)$ is the desired subgroup H. □

As an example,

$$\mathrm{Aut}(K_4 - e) \cong V_4 \leq D_8 \cong \mathrm{Aut}(\mathcal{T}(K_4 - e))$$

where V_4 is the Klein 4-group and D_8 is the dihedral group of symmetries of the square. Note that this result might not hold if G is not 2-connected. For example, let G be the star graph on five vertices with an edge added between two of the leaves; see Figure 22. The automorphism σ of G that swaps the two leaves 1 and 2 has the same effect on all of G's trees as the identity automorphism i; $\hat{\sigma}(T) = \hat{i}(T)$ for all $T \in V(\mathcal{T}(G))$. However, $\sigma(1) = 2 \neq 1 = i(1)$ when acting on $V(G)$, so $\sigma \neq i$ in $\mathrm{Aut}(G)$.

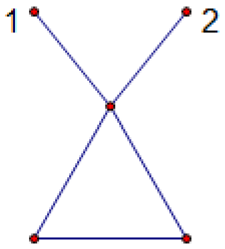


Figure 22: A non-2-connected graph that fails to satisfy the conclusion of Theorem 4.15

In general, suppose two isomorphic trees are connected to a graph G at the same vertex. Let G' be this new graph. Let γ be the automorphism of G' that swaps the two trees and fixes everything else. Then $\hat{\gamma}(T) = T$ for all $T \in V(\mathcal{T}(G'))$, since G is fixed by γ. We then have that $\hat{\gamma} = \hat{i}$ but $\gamma \neq i$. In this case Ψ would not be injective, which is what we want.

One way to prevent this issue is to assume that G is 2-connected, and thus has no cut edges. Since that is our running assumption in this paper, we do not include that hypothesis in the theorem.

If G contains a cycle, the result depends on whether or not G has any non-identity automorphisms that fix all of its cycle edges. For example, $K_3 \bigodot K_3$ is not 2-connected, but none of its seven nonidentity automorphisms fix all of its cycle edges. Thus the theorem holds for it.

Theorem 4.16. *Suppose G is 3-connected with m edges and n vertices. Then* $\mathrm{Aut}(\mathcal{T}(G)) \cong \mathrm{Aut}(G)$*, except that if* $m = 2(n-1)$*, it is also possible that* $\mathrm{Aut}(\mathcal{T}(G)) \cong \mathrm{Aut}(G) \times \mathbb{Z}_2$*.*

Proof. The cycle automorphism group, $Aut_c(G)$ is defined as the group of all functions $\phi : E(G) \to E(G)$ such that $X \subseteq E(G)$ is a cycle if and only if $\phi(X)$ is a cycle. On page 329 of [31] we have that if G is 3-connected, then $\mathrm{Aut}(G) \cong Aut_c(G)$. Let $M_G = (E, \mathcal{B})$ be the graphic matroid of G. On the same page we are told that $\mathrm{Aut}(M_G) \cong Aut_c(G)$ for any graph. Corollary 3.5 from [21] gives us that $\mathrm{Aut}(\mathcal{T}(G)) \cong \mathrm{Aut}(M_G)$, except that if $m = 2(n-1)$, it is also possible that $\mathrm{Aut}(\mathcal{T}(G)) \cong \mathrm{Aut}(M_G) \times \mathbb{Z}_2$. Following the chain of isomorphisms gives us our result. □

As an example of this result, K_5 is 3-connected, and

$$\mathrm{Aut}(\mathcal{T}(K_5)) \cong S_5 \cong \mathrm{Aut}(K_5).$$

An example of the exception is K_4, with four vertices and six edges. Indeed,

$$\mathrm{Aut}(\mathcal{T}(K_4)) \cong S_4 \times \mathbb{Z}_2 \cong \mathrm{Aut}(K_4) \times \mathbb{Z}_2.$$

A negative exception is $K_{4,3}$. It is 3-connected and has seven vertices and twelve edges, yet

$$g(K_{4,3}) = g(\mathcal{T}(K_{4,3})) = 144.$$

4.6 Induced Subgraphs and Planarity

Recall that $G - e$ is the graph where the edge e is deleted and $G \cdot e$ is the graph where the edge has been contracted. Let us use the example of the house graph H and let e be its bottom edge. Figure 23 shows H on the left with edge e in bold, $H - e$ in the middle, and $H \cdot e$ on the right. The graph $H - e$ is a triangle with two pendant vertices, which we know from Corollary 3.16 will have the same tree graph as K_3. The graph $H \cdot e$ is the same as $K_4 - e$, and we saw what its tree graph looks like in Section 2.

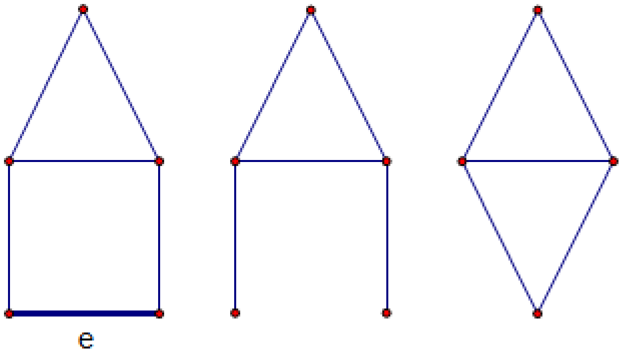


Figure 23: The house graph showing deletion and contraction of an edge

In this section we learn that tree graphs contain the tree graphs of smaller graphs inside of them. We can use this knowledge to show that essentially all tree graphs are nonplanar.

Lemma 4.17. *Let e be an edge of G. The subgraph of $\mathcal{T}(G)$ induced on all trees that do not contain e is isomorphic to $\mathcal{T}(G - e)$.*

Proof. Delete all of the trees that do contain e from $\mathcal{T}(G)$. The remaining trees will not contain e. These are exactly the spanning trees of $G - e$, with the same adjacency relationship as before. Thus what remains is $\mathcal{T}(G - e)$. □

Lemma 4.18. *Let e be an edge of G. The subgraph of $\mathcal{T}(G)$ induced on all trees that contain e is isomorphic to $\mathcal{T}(G \cdot e)$.*

Proof. Let T be a tree that contains e. If we contract e, T becomes a tree T', a spanning tree for $G \cdot e$. Likewise, for every tree $T' \in G \cdot e$, there is a spanning tree T of G that can be made by expanding ("un-contracting") e. Indeed, the two processes are inverses of each other. □

These results were found independently in earlier papers, for example Lemma 2.3 in [16]. Let us see an example of these results in action, using the same house graph H as before. Figure 24 shows $\mathcal{T}(H)$ on the left and again on the right with the induced subgraphs differentiated. The bold edges belong to $\mathcal{T}(H - e) \cong K_3$, while the dashed edges belong to $\mathcal{T}(H \cdot e) \cong \mathcal{T}(K_4 - e)$. Notice that each vertex in the graph is in exactly one of these two induced subgraphs. The ones in the triangle are all of the spanning trees that do not contain e, and the remaining vertices are the trees that do contain e. Cummins makes use of this deletion/contraction property in his original paper [4].

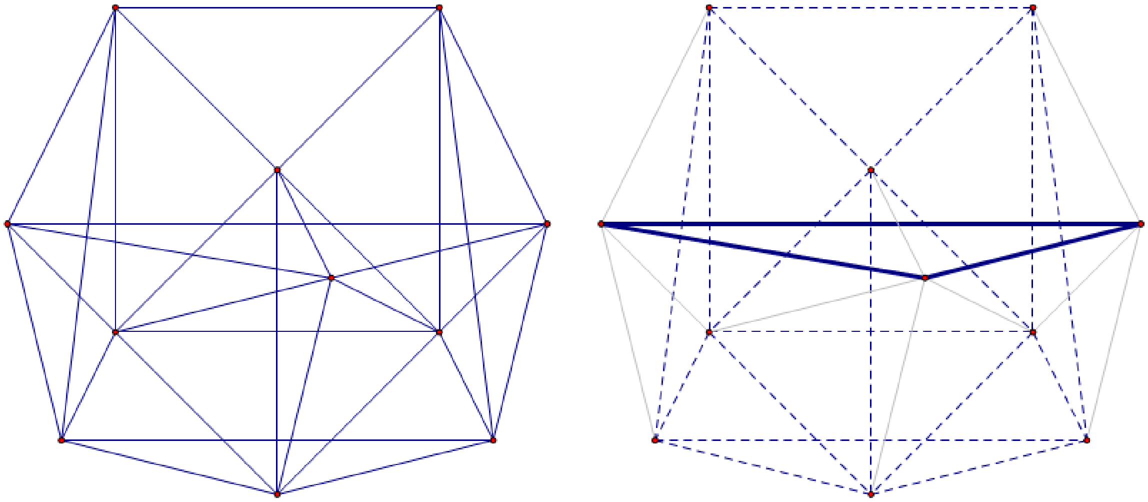


Figure 24: $\mathcal{T}(H)$ with the noted induced subgraphs shown

H is a *minor* of a graph G, written $H \sqsubseteq G$, if we can obtain H from G by a sequence of deleting vertices, deleting edges, and contracting edges. Minors are generalizations of subgraphs. For instance, $C_4 \sqsubseteq C_5$, but $C_4 \not\leq C_5$. Wagner's theorem [30], following Kuratowski's celebrated work, says that a graph is planar if and only if it does not contain K_5 or $K_{3,3}$ as a minor. The following corollary summarizes the findings of this section so far.

Corollary 4.19. *If H is a minor of G then $\mathcal{T}(H)$ will be an induced subgraph of $\mathcal{T}(G)$.*

In diagram form, Corollary 4.19 says:

$$\begin{tikzcd}
\mathcal{T}(H) \arrow[r, phantom, "\leq"] & \mathcal{T}(G) \\
H \arrow[u] \arrow[r, phantom, "\sqsubseteq"] & G \arrow[u]
\end{tikzcd}$$

We can use this relationship to investigate the planarity of tree graphs. Our tactic will be to show that two small tree graphs are not planar and that almost every graph has one of those two base graphs as a minor, causing their tree graphs to also be nonplanar.

Lemma 4.20. *The tree graphs $\mathcal{T}(K_4 - e)$ and $\mathcal{T}(K_3 \bigodot K_3)$ are not planar.*

Proof. The graph $K_4 - e$, also known as the *diamond*, and the joined triangle graph $K_3 \bigodot K_3$ from Figure 9, also called the *butterfly*, play a special role in deciding the planarity of tree graphs. The tree graph of the diamond is not planar since it contains a K_5 minor. Likewise, the tree graph of the butterfly is not planar as it contains $K_{3,3}$ as a minor. The next two figures demonstrate this. The first transformation is shown in Figure 25. First the bottom left vertex is deleted, then the dashed edges are contracted to reveal K_5. The left graphic in Figure 26 shows $\mathcal{T}$(butterfly). The middle graphic is a redrawing of it after deleting some edges. Once the dashed edges are contracted, we get $K_{3,3}$ on the right.

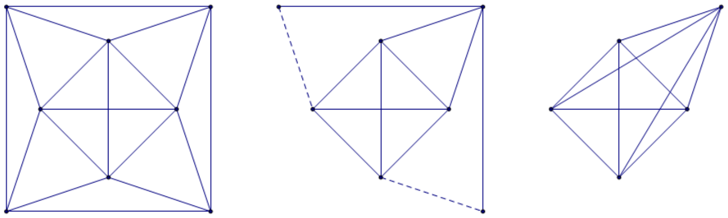


Figure 25: Showing the tree graph of the diamond is nonplanar

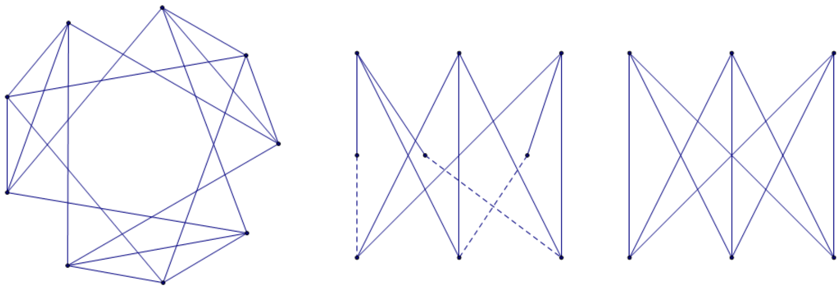


Figure 26: Showing the tree graph of the butterfly is nonplanar

□

Theorem 4.21. *The tree graph $\mathcal{T}(G)$ is nonplanar unless $G \cong C_3$ or $G \cong C_4$.*

Proof. By Corollary 4.19 and Lemma 4.20 we know that the tree graph of any graph containing the diamond or the butterfly as a minor will be nonplanar. So which graphs do not have these as minors? If we forbid the diamond as a minor, [6] tells us that we are left with the *cactus* graphs, which are connected graphs in which any two cycles have at most one vertex in common. By also forbidding the diamond as a minor, we leave ourselves with the 2-connected cacti. Lemma 2 in [6] tells us that only cycles remain.

By Lemma 3.8 we know that tree graphs of cycles are complete graphs. Any complete graph K_n, for $n \geq 5$ will contain K_5 as a subgraph, and so will be nonplanar. Thus $\mathcal{T}(C_n)$ is nonplanar for $n \geq 5$. By inspection we know that K_3 and K_4 are planar, so $\mathcal{T}(C_3)$ and $\mathcal{T}(C_4)$ are planar. Those are the lucky two. □

4.7 Clique Decomposition

A *graph decomposition* is a partition of the edges of a graph. Essentially, we are breaking down the graph into smaller component parts. Sometimes we would like these parts to all share a particular property. One example of this is a *clique decomposition*, in which the edges in every cell of the partition form a complete subgraph in the given graph. This can be done trivially, as technically a single edge is the same as the complete graph K_2. Thus clique decompositions are usually restricted to cliques of order three and larger.

Not all graphs have such a clique decomposition. Consider the diamond, for example. It does not have enough edges to be a K_4, and pulling out a K_3 leaves two edges behind. In this section we will show that every tree graph has a clique decomposition. Moreover, every vertex will be in the same number of cliques. This is a very restrictive condition to place on graphs, one which might help to characterize tree graphs.

In a graph with n vertices and m edges, let $v = m - n + 1$. This is a measure of the number of "extra" edges of G, as the number of edges leftover whenever a spanning tree is taken out.

Theorem 4.22. *Let G be a graph with n vertices and m edges, and let $v = m - n + 1$. The edges of $\mathcal{T}(G)$ can be decomposed into cliques of size at least three such that each vertex is in exactly v cliques.*

Proof. Start with any vertex of $\mathcal{T}(G)$, which corresponds to a spanning tree of G. To walk from it to its neighbors in $\mathcal{T}(G)$, we first must add an edge to it, creating a unique cycle of size at least three. By Lemma 2.4, removing any other edge from this cycle gives us a new spanning tree, whose corresponding vertex must be adjacent to the vertex for our starting tree in $\mathcal{T}(G)$. Adding that extra edge creates a unicycle. Each spanning unicycle with cycle size $c \geq 3$ in G gives rise to a K_c clique in $\mathcal{T}(G)$ (see Lemmas 3.8 and 3.16 and Corollary 4.19). This tells us that each spanning unicycle that contains a given tree produces a unique clique in $\mathcal{T}(G)$. How many such unicycles contain a given tree? Well there are v extra edges in G that can be added in order to create the unicycles. Therefore every tree is part of v such unicycles, and every vertex in $\mathcal{T}(G)$ is part of v cliques. □

The size of a clique that contains a given edge in $\mathcal{T}(G)$ can be found by looking at the union of the trees that are incident to it and finding the size of the cycle that is created. Figure 27 shows such a clique decomposition for $\mathcal{T}(G)$, where G is the diamond graph. There is a K_4 in bold and four K_3 subgraphs in thin and dashed edges. We can see why the cliques arise by looking at the spanning trees of $K_4 - e$ in Figure 3. The four interior trees do not have the diagonal edge, so the union of any two of them is C_4. The four exterior trees do contain the diagonal edge, so any union involving them will give a C_3 which includes that edge.

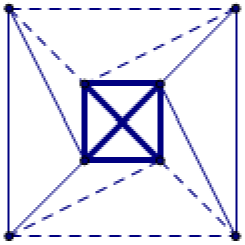


Figure 27: Clique decomposition of the tree graph of the diamond

We can use this decomposition property to predict the number of edges in $\mathcal{T}(G)$. To do so, we just need to find all of the spanning unicycles of G and record the size of the cycle in each of them.

Corollary 4.23. *Let G be any graph and let $u_1, u_2, \ldots, u_\mu$ be all of the distinct spanning unicycles of G. Let $c_1, c_2, \ldots, c_\mu$ be the list of the sizes of the unique cycles contained in these unicycles. Then*

$$|E(\mathcal{T}(G))| = \sum_{i=1}^{\mu} \binom{c_i}{2}.$$

Proof. Theorem 4.22 gives us that $\mathcal{T}(G)$ can be decomposed into cliques and that each $k-$clique arises from a $k-$cycle contained in a unicycle. Once we know the sizes of the cycles in all of the unicycles, we know the kinds of cliques in the decomposition of $\mathcal{T}(G)$. Since K_n has $\binom{n}{2}$ edges, we get the total number of edges by simply adding up all of the appropriate binomial coefficients. □

As an example, consider the diamond. Figure 28 shows its five spanning unicycles. The first four contain C_3, so will contribute $4 \cdot \binom{3}{2} = 12$ edges. The last contains C_4, so adds $\binom{4}{2} = 6$ edges, for a total of 18. This matches what we see in Figure 27.

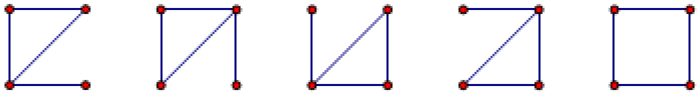


Figure 28: The $\mu = 5$ unicycles of the diamond

Recall that a graph is bipartite if its vertices can be partitioned into two sets such that every edge in the graph has one endpoint in each set.

Corollary 4.24. *For any 2-connected graph G, the tree graph $\mathcal{T}(G)$ is not bipartite.*

Proof. Since G is 2-connected, it contains a cycle. From Corollary 4.19 we know that cycles in graphs give rise to complete graphs in their tree graphs. Each such complete graph contains at least one triangle, so $\mathcal{T}(G)$ contains at least one triangle. Therefore $\mathcal{T}(G)$ has at least one odd cycle, and so by König's theorem [32, Theorem 1.2.18], $\mathcal{T}(G)$ cannot be bipartite. Additionally, we get that $\text{girth}(\mathcal{T}(G)) = 3$ for all such G. □

4.8 Special Families

When investigating tree graphs, several special families stand out. One of those is the family of *theta graphs*. The graph $\theta_{a,b,c}$ is made by joining two vertices with openly-disjoint paths of length a, b, and c, with at least two of the values being greater than one. For example, the house graph is isomorphic to $\theta_{3,2,1}$. Simple counting shows us that $\theta_{a,b,c}$ has $a + b + c$ edges and $a + b + c - 1$ vertices. This means that no matter what the parameters are, $v = m - n + 1 = 2$ for theta graphs.

Line graphs were mentioned in the introduction as another example of a deterministic graph construction that yields interesting results. The line graph, $L(G)$, has a vertex for every edge of G, where two vertices are adjacent if and only if their corresponding edges are incident in G. Thanks to Krausz's characterization of line graphs

[12], we know that a graph is a line graph if and only if its edges can be partitioned into cliques such that each vertex is in at most two cliques (here cliques can have size smaller than three). We can use this strong property to describe tree graphs that are also line graphs. The next two theorems do just that.

Theorem 4.25. *For any integers $a \geq b \geq c \geq 1$, with $b \geq 2$,*

$$\mathcal{T}(\theta_{a,b,c}) \cong L(K_{a,b,c}).$$

Proof. The number of vertices of $\mathcal{T}(\theta_{a,b,c})$ is the number of spanning trees of $\theta_{a,b,c}$. Since $v = 2$ and $\theta_{a,b,c}$ has three cycles, we have to remove two edges and break three cycles to get a spanning tree. We cannot remove two edges from just one of the three disjoint paths, as the union of the other two would still be a cycle. Thus we must remove one edge from two distinct paths to create a tree. There are $ab + ac + bc$ ways to do this, which tells us the number of vertices of $\mathcal{T}(\theta_{a,b,c})$.

The number of vertices of $L(K_{a,b,c})$ is the number of edges in $K_{a,b,c}$. Each vertex in a cell is adjacent to exactly all of the vertices in the other two cells. Thus there are

$$\begin{aligned}\frac{a(b+c) + b(a+c) + c(a+b)}{2} &= \frac{ab + ac + ba + bc + ca + cb}{2} \\ &= \frac{2ab + 2ac + 2bc}{2} \\ &= ab + ac + bc\end{aligned}$$

edges in $K_{a,b,c}$, which gives us the number of vertices in $L(K_{a,b,c})$.

We have now seen that the two graphs have the same number of vertices. Label each edge of $\theta_{a,b,c}$ with the numbers 1 through $a + b + c$. By construction, each of the three paths in $\theta_{a,b,c}$ has a corresponding cell in $K_{a,b,c}$ of the same size. Label the vertices of $K_{a,b,c}$ so that the vertices in the cells get the same labels as the edges in their corresponding paths. In this way we can set up a bijection between the trees of $\theta_{a,b,c}$ and the edges of $K_{a,b,c}$. Each tree can be identified by the labels of the two edges that were removed in order to create it. We then identify tree $\{i,j\}$ with edge ij (see Figure 29). This gives us a bijection between the vertices of $\mathcal{T}(\theta_{a,b,c})$ and the vertices of $L(K_{a,b,c})$.

Two trees in $\mathcal{T}(\theta_{a,b,c})$ are adjacent if they differ by a single edge — that is, if their labels differ in a single place. Thus $\{i,j\}$ will be adjacent to $\{i,k\}$ and $\{k,j\}$ for all available values of k. Adjacency in $L(K_{a,b,c})$ is determined by edges that are incident to each other — that is, if they share a vertex. So edge ij will be incident to edges ik and kj for all available values of k. Thus the adjacency relationship is preserved by the bijection and therefore the two graphs are isomorphic. □

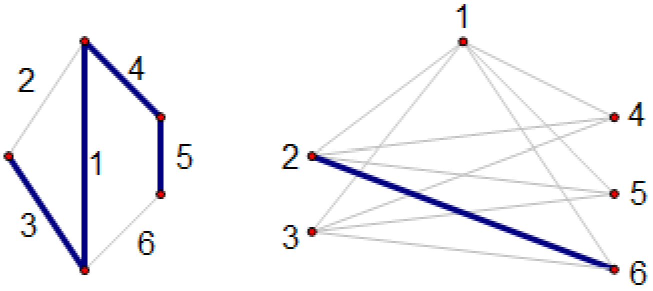


Figure 29: Spanning tree $\{2,6\}$ of $\theta_{3,2,1}$ and corresponding edge $\{2,6\}$ in $K_{3,2,1}$

Theorem 4.26. *Let G be a 2-connected graph. There are at least two possibilities for when $\mathcal{T}(G)$ is a line graph, i.e. $\mathcal{T}(G) \cong L(H)$ for some graph H:*

1. *$G \cong C_n$ and $H \cong K_{1,n}$.*

2. *$G \cong \theta_{a,b,c}$ and $H \cong K_{a,b,c}$.*

Proof. Theorem 4.22 tells us that each vertex of $\mathcal{T}(G)$ is partitioned into $v = m-n+1$ cliques of size at least three. So by Krausz, we consider the cases when $v = 1$ or $v = 2$.

Case $v = 1$: In this case, G has the same number of vertices and edges. For 2-connected graphs, this only leaves the cycles. Lemma 3.8 gives us that tree graphs of cycles are complete graphs, and it is not hard to see that $K_n \cong L(K_{1,n})$.

Case $v = 2$: In this case, G has one more edge than it has vertices. We know G is 2-connected. Let $\Psi(v)$ be the maximum number of cycles in a graph with parameter v. In [7] we learn that $2^{v-1} + v^2 - 3v + 3 \leq \Psi(v) \leq 2^v - 1$. Plugging in $v = 2$, we get $3 \leq \Psi(2) \leq 3$, which tells us that G can have at most three cycles. It cannot have zero cycles since $m > n$, nor can it have exactly one cycle since $m \neq n$. Thus it either has two or three cycles. If it has exactly two cycles, they must not share an edge, otherwise a third cycle would be created from their boundary. Likewise there cannot be more than one path between the two cycles, otherwise an additional cycle would be created. Then either the cycles share a vertex or there is a single path joining them. In either case, the graph has a cut-vertex and so is not 2-connected. Thus there are exactly three cycles in G. This is only possible if G is a theta graph.

Thus $G \cong \theta_{a,b,c}$ as described above. By Theorem 4.25, $\mathcal{T}(\theta_{a,b,c}) \cong L(K_{a,b,c})$. This finishes the result. □

For example, since the house graph is isomorphic to $\theta_{1,2,3}$, we have that $\mathcal{T}(\text{house}) \cong L(K_{1,2,3})$.

Because of this relationship between theta graphs and complete tripartite graphs via the line graph, we can say something about the automorphism group of their tree graphs. Recall that the *glory* of a graph G, $g(G)$, is the size of its automorphism group.

Theorem 4.27. *For any integers $a \geq b \geq c \geq 1$, with $b \geq 2$, the glory*

$$g(\mathcal{T}(\theta_{a,b,c})) = s!\,a!\,b!\,c!$$

where s is the number of parameters among $\{a, b, c\}$ that are equal.

Proof. From Theorem 4.25 we know that $\mathcal{T}(\theta_{a,b,c}) \cong L(K_{a,b,c})$. Thus we have that $\mathrm{Aut}(\mathcal{T}(\theta_{a,b,c})) \cong \mathrm{Aut}(L(K_{a,b,c}))$. Corollary 1.4 in [15] says that $\mathrm{Aut}(G) \cong \mathrm{Aut}(L(G))$ under conditions which our complete tripartite graphs meet. Therefore $\mathrm{Aut}(\mathcal{T}(\theta_{a,b,c}))$ $\cong$ $\mathrm{Aut}(K_{a,b,c})$. Since $\mathrm{Aut}(K_{a,b,c}) \cong (S_a \times S_b \times S_c) \rtimes S_s$, we have $g(\mathcal{T}(\theta_{a,b,c})) = s!\,a!\,b!\,c!$. □

For example, $g(\mathcal{T}(\theta_{2,2,2})) = 6 \cdot 2^3 = 48$, and $g(\mathcal{T}(\theta_{4,2,2})) = 2 \cdot 24 \cdot 2^2 = 192$, while $g(\mathcal{T}(\theta_{3,2,1})) = 1 \cdot 6 \cdot 2 \cdot 1 = 12$.

We can combine earlier results to get tight bounds on the chromatic number of tree graphs of theta graphs.

Theorem 4.28. *Let $a \geq b \geq c \geq 1$, with $b \geq 2$. Then*

$$a + b \leq \chi(\mathcal{T}(\theta_{a,b,c})) \leq a + b + c.$$

Proof. At the beginning of Section 4.8 we saw that $\theta_{a,b,c}$ has $a + b + c - 1$ vertices and $a+b+c$ edges. Since $\theta_{a,b,c}$ has exactly three cycles, one each of sizes $a+b, a+c$, and $b+c$, finding the circumference of $\theta_{a,b,c}$ amounts to choosing the greatest of those values. By our ordered labeling of the parameters, this is $a + b$. Corollary 3.23 then gives us $a+b = \mathrm{circ}(\theta_{a,b,c}) \leq \chi(\mathcal{T}(\theta_{a,b,c}))$. Further, $\chi(\mathcal{T}(\theta_{a,b,c})) \leq |E(\theta_{a,b,c})| = a+b+c$ by Theorem 3.18. Putting these bounds together gives the result. □

When $c = 1$, the graph, $\theta_{a,b,c}$ is hamiltonian and the bounds only differ by one. Data gathered suggests that the lower bound is usually the correct value.

The next nice family of base graphs is the $P_{n,k}$ family. Recall that $P_{n,k}$ is the graph formed by joining two vertices with n openly-disjoint paths of length k. The graph $P_{n,k}$ has nk^{n-1} spanning trees. To see this, notice that to avoid cycles, all but one of the paths must be broken by removing a single edge, and that removing more than one edge from any path would disconnect the graph. There are n choices for the unbroken path. There are $n - 1$ paths left to break, and k edges in each path from which to select one for removal. This gives us nk^{n-1} spanning trees, which also tells us the number of vertices of $\mathcal{T}(P_{n,k})$.

The tree graphs of this family are regular and are extremely glorious. They also seem to be integral and vertex transitive, topics which will be explored in the next section. Additionally, it can be shown that $\mathcal{T}(P_{n,2})$ is related to the $n-$dimensional hypercube. (Recall that the hypercube Q_n is defined as K_2^n, where the exponentiation is taken over the Cartesian product.)

The construction works as follows. Start with the hypercube Q_n. Next, take its line graph. Then connect two vertices in the line graph if their respective edges were opposite each other on a 4-cycle face in Q_n. Figure 30 illustrates this process when $n = 3$. In (1) we start with Q_3. The image in (2) adds $L(Q_3)$. In (3) we add the extra edges across all of the square faces. Finally in (4) we see the completed $\mathcal{T}(K_{3,2})$ by itself. As we do not rely on this result, we omit the verification of this construction.

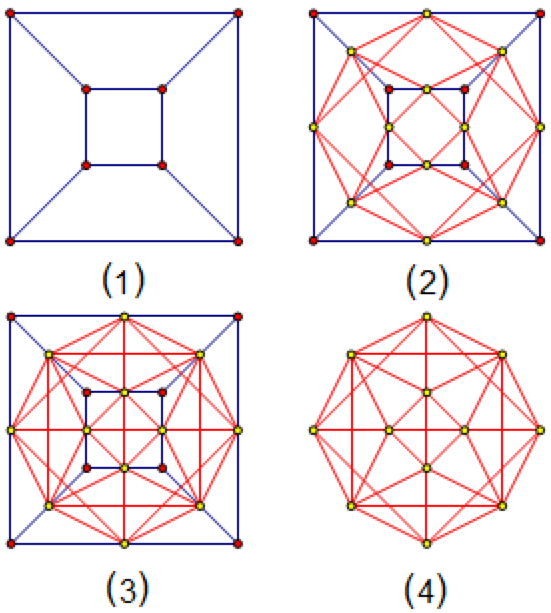


Figure 30: Constructing $\mathcal{T}(K_{3,2})$ from Q_3

Theorem 4.29. *The graph $\mathcal{T}(P_{n,k})$ is regular with degree $(n-1)(2k-1)$.*

Proof. As discussed above, each tree of $P_{n,k}$ is missing $n-1$ edges: one each from $n-1$ of the paths. Picking any one of these to add back to the graph creates a single $2k-$cycle. Removing any of the $2k$ edges in this cycle other than the one just added will create a new tree adjacent to the first in $\mathcal{T}(P_{n,k})$. This gives us $(n-1)(2k-1)$ adjacent trees, and the degree of each vertex in the graph. □

As mentioned earlier, these tree graphs are highly symmetric. As n and k increase, the automorphism group grows very rapidly. For example, $g(\mathcal{T}(P_{3,3})) = 1296$, while $g(\mathcal{T}(P_{3,5})) = 10368000$.

The $\theta_{a,b,c}$ and $P_{n,k}$ graphs are similar in that they are both formed by joining two vertices with openly-disjoint paths. In the former, there are exactly three paths but their lengths can be different. So $\theta_{a,b,c}$ is a graph with exactly three cycles, which is the minimum possible among 2-connected graphs that are not cycles. In the latter, the number of paths can vary but their lengths are the same. This allows $P_{n,k}$ to have many cycles of length $2k$ (in particular, $\binom{n}{2}$ of them). In either case, the simplicity of the cycle structure of the graph leads to nice structural properties of its tree graph. We can also view even cycle graphs C_{2k} as special cases of the $P_{n,k}$ family. Having a single cycle leads to a complete tree graph, which has the most automorphisms possible.

4.9 Conjectures

While exploring the data generated in this investigation of tree graphs, many patterns emerged. Some of these patterns were nailed down and turned into the results proved elsewhere in this paper. Other patterns could not be explained as easily, and thus are (hopefully temporarily) left as conjectures. This section collects some of the more compelling conjectures put forth so far. Many of them are algebraic in nature, and all of them help build the allure of tree graphs.

Conjecture 4.30. *The clique decomposition from Theorem 4.22 is unique.*

When $v = 1$ or 2, we know from Theorem 4.26 that our tree graphs are line graphs. Krausz's characterization says that we can decompose line graphs into cliques, and Whitney's isomorphism theorem [33] says that except in the case of one counterexample (a counterexample that does not occur among tree graphs), isomorphic line graphs are equivalent to isomorphic graphs. In other words, $L(G) \cong L(H) \iff G \cong H$. Since the base graph is reconstructed from the clique decomposition of the line graph, this implies that the decomposition is unique. It might not be a stretch to assume that unique decomposition holds for all values of v, and thus all tree graphs.

Even if this is true, it still does not help us necessarily find the base graph that induces a particular tree graph. Theorems 3.9 and 3.10 are thorns in our side that prevent us from knowing if a unique graph gives rise to our tree graph. Nonetheless, it would provide useful information about the base graph, such as the possible sizes of cycles in G.

Conjecture 4.31. *Let G be a graph with $n > 3$ vertices. If* $\mathrm{Aut}(G) \cong \mathrm{Aut}(\mathcal{T}(G))$ *then G is 3-connected.*

We know the converse is true by Theorem 4.16 (unless perhaps $m = 2(n-1)$). This conjecture would suggest that the properties are equivalent for tree graphs (again, except for perhaps in the case when $m = 2(n-1)$). The contrapositive of this conjecture, along with Theorem 4.15, says that if G is 2-connected but not 3-connected, $\mathrm{Aut}(G)$ will always be a proper subgroup of $\mathrm{Aut}(\mathcal{T}(G))$. The restriction on n eliminates the pesky K_3 counterexample, which by definition is only 2-connected.

We indicated that the tree graphs of $P_{n,k}$ graphs were glorious, but exactly how glorious are they?

Conjecture 4.32. *For any integers $n, k \geq 2$, the glory of the graph $P_{n,k}$ is given by*

$$g(\mathcal{T}(P_{n,k})) = \begin{cases} (2k)!, & n = 2 \\ n!(k!)^n, & n > 2 \end{cases}$$

When $n = 2$, we have $P_{2,k}$, which is just a $2k-$cycle, and we know this has the tree graph K_{2k}. So the automorphism group is S_{2k}, which tells us $g(\mathcal{T}(P_{2,k})) = (2k)!$. Otherwise, the automorphism group seems to be the wreath product $S_k \wr S_n$. When $k = 2$ we get the special case of $S_2 \wr S_n$, which are the *hyperoctahedral* groups.

Conjecture 4.33. *Let G be a graph with $n > 3$. If G is 2-connected but not 3-connected, then $g(\mathcal{T}(G))$ is divisible by 4.*

The connectivity restriction of Conjecture 4.33 is based on an extension of Frucht's theorem [8], which says that for any group Γ there is a graph G whose automorphism group realizes Γ. We can in fact find such a graph G that is as connected as we want [27]. Thus we can pick any number n, construct a 3-connected graph which has that glory, say with automorphism group $\mathbb{Z}_n$, and by Theorem 4.16 we get $g(\mathcal{T}(G)) = n$.

The triangle K_3 has the symmetry group of the dihedral group on six elements, and $\text{Aut}(K_3) \cong \text{Aut}(\mathcal{T}(K_3))$ so $g(\mathcal{T}(K_3)) = 6$. This has been the only 2- but not 3-connected graph found so far whose glory was not divisible by four. One way Conjecture 4.33 could be true is if every automorphism group for this type of tree graph contains a subgroup of size four.

Conjecture 4.34. *For any graph G, the ratio $g(\mathcal{T}(G))/g(G)$ is 1, 3, or even.*

Theorem 4.15, along with Lagrange's theorem, says the given ratio has to be a whole number. By Theorem 4.16 the ratio is one when G is 3-connected (Conjecture 4.31 says this is the only time). When $G = C_4$, we have $\text{Aut}(G) \cong D_8$ while $\text{Aut}(\mathcal{T}(G)) \cong S_4$, which gives a glory ratio of three. So far, C_4 has been the only graph found to have this ratio, and besides one and three, the rest of the ratios have been even.

The *adjacency matrix* $A(G)$ of a graph G is a square matrix with rows and columns indexed by the vertices of G, where an entry is 1 if the corresponding pair of vertices is adjacent, and 0 otherwise. The eigenvalues of $A(G)$ can tell us useful information about the graph [5, 9]. If all of the eigenvalues of $A(G)$ are integers, G is called an *integral* graph. If every vertex of G can be sent to every other vertex by means of an automorphism, then G is called *vertex-transitive.*

Conjecture 4.35. *For any integers $n, k \geq 2$, the graph $\mathcal{T}(P_{n,k})$ is integral and vertex-transitive.*

The adjacency matrix $A(\mathcal{T}(P_{n,k}))$ seems to have $\lceil \frac{3n-2}{2} \rceil$ distinct eigenvalues. Since it is regular, the maximum eigenvalue will be $(n-1)(2k-1)$. Then there are $\lceil \frac{n-1}{2} \rceil$ jumps of size $2k$ between the next biggest distinct eigenvalues, then jumps of size k down to the smallest eigenvalue of $1-n$.

Figure 31 shows $P_{3,2}$ on the left next to its tree graph. In this example, the list of eigenvalues and their multiplicities of $\mathcal{T}(P_{3,2})$ is $6, 2^{(3)}, 0^{(2)}, -2^{(6)}$. See Appendix C for more information.

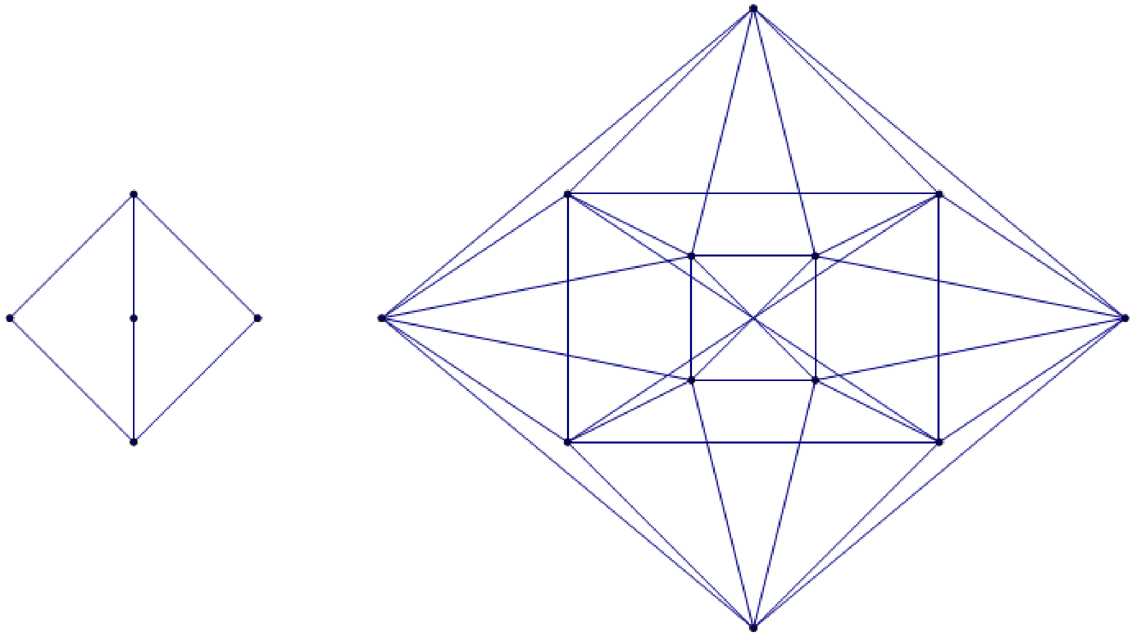


Figure 31: $P_{3,2}$ and its tree graph

We now know many useful properties of $\mathcal{T}(P_{n,k})$ and can use them to construct a formidable graph with quite a bit of freedom of choice. Suppose you wanted a regular graph; then $\mathcal{T}(P_{n,k})$ will not disappoint. Perhaps you are picky and want to choose the degree? We saw in Theorem 4.8 that $\mathcal{T}(P_{n,k})$ is $(n-1)(2k-1)-$regular, and $(n-1)$ can be be any positive integer we want. Since $2k-1$ is odd and at least three, their product gives us the ability to generate any natural number that is not a power of two (we can get those as well, if G is allowed to be a non-simple graph). For example, say you want a 60-regular graph. Then any of the graphs $\mathcal{T}(P_{5,8}), \mathcal{T}(P_{13,3})$, or $\mathcal{T}(P_{21,2})$ will satisfy you. Did you also want your graph to be integral? How about vertex-transitive? Maximally-connected? All of the above? No problem. Even hamiltonian-connected is guaranteed! These are some very nice graphs. Figure 32 shows a drawing of $\mathcal{T}(P_{4,2})$.

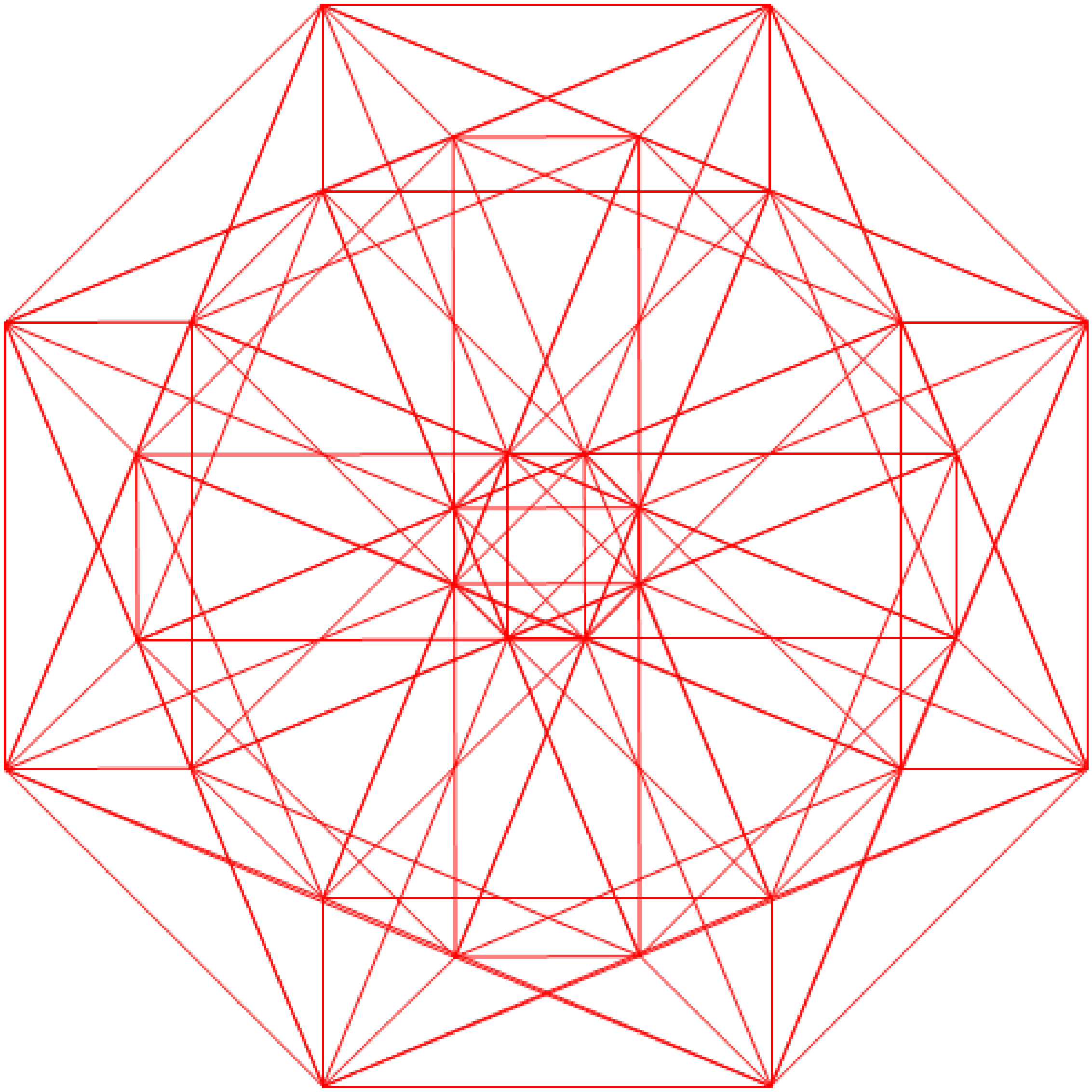


Figure 32: The tree graph $\mathcal{T}(P_{4,2})$

Patterns were also found in the multiplicities of eigenvalues of $\mathcal{T}(P_{n,2})$. See Appendix C for more information.

The tree graphs of theta graphs seem to be almost integral. Let $a, b,$ and c be distinct. The data collected suggests that $\mathcal{T}(\theta_{a,b,c})$ has only three non-integer eigenvalues while $\mathcal{T}(\theta_{a,b,b})$ has only two. For $\mathcal{T}(\theta_{a,b,b})$, the integer eigenvalues seem to be

$$\{-2, a-2, 2(b-1), a+b-2\}$$

while the two irrational eigenvalues are

$$\{.5(a+4(b-1) \pm \sqrt{a(a+8b)})\}.$$

Certain values of a and b make these last two integers, such as $(a, b) = (1, 10)$ and $(2, 6)$.

Let $r = a+b+c$. The non-integer eigenvalues of $\mathcal{T}(\theta_{a,b,c})$ seem to be the roots of the monic cubic polynomial

$$x^3 - 2(r-3)x^2 + (r-6)(r-2)x + d,$$

where d is some constant mysteriously dependent on $a, b,$ and c. For example, the triple $(5, 3, 1)$ gives the cubic $x^3 - 12x^2 + 21x + 38$, while for $(4, 3, 2)$, which has the same r value of 9, we get $x^3 - 12x^2 + 21x + 2$.

The only regular tree graphs so far have been when the base graph is a cycle, $P_{n,k}$, or a *bracelet* — copies of one of the two former types connected together in a cycle like a bracelet. Figure 33 shows bracelets made from three copies each of C_4 on the left and $P_{3,2}$ on the right. Their tree graphs are 16-regular and 27-regular, respectively. The regularity seems to come from the fact that there are only two types of cycles present: in these examples, only C_4 and C_6. Also of importance might be the fact that in all three of these families, any two cycles share at least one vertex.

A counting argument shows that if b copies of $P_{n,k}$ are arranged to make a bracelet graph B, then B is regular with degree $(bk-1)n + (2k-1)(b-1)(n-1)$.

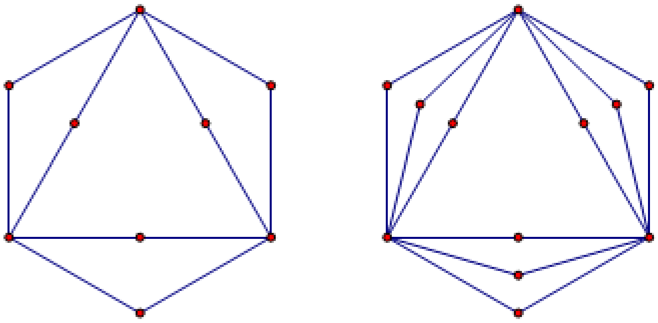


Figure 33: Two bracelet graphs with regular tree graphs

Conjecture 4.36. *If the repeated graph in a bracelet B has a regular tree graph, then $\mathcal{T}(B)$ is regular.*

All examples generated in the direction of this conjecture have produced regular tree graphs. Conversely, trying to make bracelets out of graphs with nonregular tree graphs, such $K_4 - e$, has not yielded regular tree graphs. It might be useful to have a more formal definition of a bracelet graph built from a given graph, and to explore other bracelet-like constructions.

The degree bounds from Theorem 3.19 tell us that if the girth and circumference of a graph are the same, i.e. if the graph has only one cycle size, then its tree graph will be regular. Cycle graphs and $P_{n,k}$ graphs fall into this category. The bracelets do not have a single cycle size, however, showing that it is a sufficient but not necessary property for tree graph regularity.

It should also be noted that bracelets do not seem to inherit integrality from their base graphs. The graph $P_{n,k}$ is integral, but data suggests that the tree graph of a bracelet built from $P_{n,k}$ is not. In this way integrality seems to be a stronger condition than regularity. However, there are examples of nonregular tree graphs that are integral, such as $\mathcal{T}(\theta_{3,3,1})$.

Conjecture 4.37. *If $\mathcal{T}(G)$ is regular, it is vertex-transitive. Furthermore, it is integral.*

Vertex-transitivity implies regularity. Since automorphisms preserve the structure of a graph, vertices can be mapped only to vertices of the same degree. In a vertex-transitive graph, every vertex can be mapped to every other vertex. Thus all degrees must be the same and the graph is regular. The converse does not hold in general, however. Figure 34 shows a 3-regular graph that is not vertex-transitive. Some of the vertices are part of two four-cycles and one five-cycle, while others are part of two five-cycles and one four-cycle. There is no way to map these types to each other. Despite the failure of this implication for general graphs, we have no counterexamples (yet) among tree graphs, hence Conjecture 4.37.

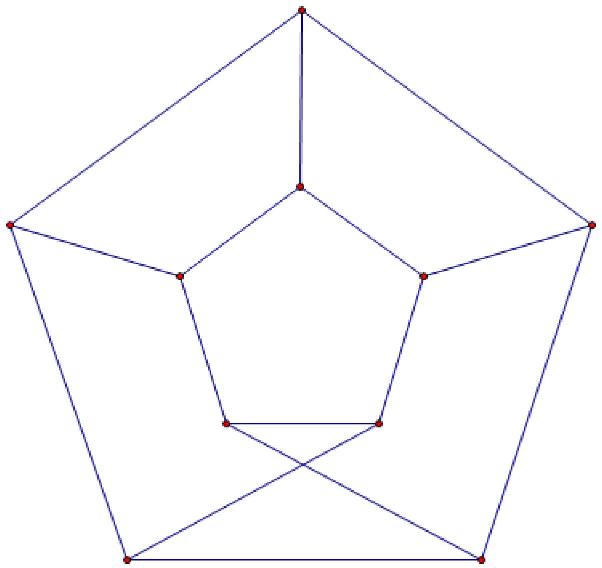


Figure 34: A regular graph that is not vertex-transitive

By Theorem 4.22 we can decompose the edges of $\mathcal{T}(G)$ into cliques of at least size three such that each vertex is part of the same number of cliques. Consider a clique decomposition of a tree graph. The *partition type* of a vertex with reference to a clique decomposition is the multiset of clique sizes that contain the vertex. For example, the only two partition types of the vertices in the decomposition of $\mathcal{T}$(diamond), seen in Figure 27, are $\{3,3\}$ and $\{3,4\}$. Only regular graphs can be vertex-transitive, but there is a generalized version of this property that all tree graphs seem to share. In this case, the graph is vertex-transitive on all vertices of the same partition type. More formally, we have the following.

Conjecture 4.38. *For each partition type p of G, the subgraph induced on all vertices of partition type p is vertex transitive.*

Let us take a look at a few examples. The subgraphs induced on the partition types of $\mathcal{T}$(diamond) are C_4 and K_4. For $\mathcal{T}$(house) we get K_2, K_3, and $\text{Prism}_3 \cong K_2 \square K_3$. With $\mathcal{T}(K_{1,1,3})$ we get $\mathcal{T}(K_{3,2})$ and Q_3. For all of these graphs, unique degrees correspond to unique partition types. That is not always the case, however. Let H' be the graph depicted in Figure 35. Then $\mathcal{T}(H')$ has ten vertices of degree 9. Eight of them have partition type $\{6,3,3\}$ while the remaining two have partition type $\{5,4,3\}$. The subgraph induced on the former type is Q_3, while the latter type induces K_2. All of these induced subgraphs are vertex transitive.

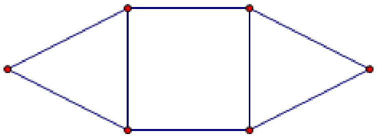


Figure 35: A graph whose tree graph shows degree and partition type are not equivalent

If Conjecture 4.38 is true, we may be able to say something about the automorphism group of a tree graph based on the automorphism groups of these vertex-transitive subgraphs. A partition type of size k would need at least k automorphisms to be transitive on that set of vertices. So

$$g(\mathcal{T}(G)) \geq \max\{k \mid p \text{ is a partition type of size } k \text{ in } G\}.$$

One lofty goal would be to characterize tree graphs. That is, given a graph G, we want to be able to know for sure whether or not there exists a graph H such that $G \cong \mathcal{T}(H)$. We saw in Section 4.3 that all tree graphs are matroid basis graphs. Maurer came up with several characterizations of matroid basis graphs in [20]. However, these characterizations involve many properties that are tough to check in practice. Since tree graphs constitute a proper subclass of the class of matroid basis graphs that behaves quite nicely, the hope is that there is a nicer way to characterize them. For now we simply tack on a few necessary conditions that seem strong enough and hope for the best.

Conjecture 4.39. *For any graph G, $G \cong \mathcal{T}(H)$ for some graph H if and only if G is a matroid basis graph with no induced octahedra and can be decomposed into cliques of size three or more, where each vertex is in the same number of cliques.*

Theorem 4.10 and Theorem 4.22 give us that these are necessary conditions. This conjecture posits that they are also sufficient conditions. It has been difficult so far to try to produce a graph meeting these conditions that is not already known to be a tree graph, since the conditions are hard to check. We hope that this conjecture could lead to a classification of tree graphs, although even if it is successful it will be of more theoretical than practical use.

5 Trees and Decompositions of Complete Graphs

5.1 Background and Terminology

A *matching* in a graph is a set of disjoint edges. A *perfect matching* in a graph with an even number of vertices is a matching that includes every vertex. A perfect matching is also called a *1-factor*. A *1-factorization* of a graph is a set of perfect matchings that partitions the edge set.

As it happens, there are many different 1-factorizations of the complete graph on an even number of vertices, K_{2n}. Categorizing these has led to a number of interesting results, including the designation of several infinite families. One such family is GK_{2n}. To describe this family, we begin by placing $2n-1$ vertices evenly-spaced in a circular arrangement around a single, central vertex. Next, a perfect matching is formed by adding an edge that joins the central vertex to one of the outer vertices, and then drawing in all of the edges that are perpendicular to the first one. To complete the 1-factorization, we form the remaining perfect matchings by simply rotating this design around the central vertex. An illustration of this construction is shown in Figure 36. The first two matchings are shown, and then the complete 1-factorization is given, with matchings distinguished by the colors of the edges.

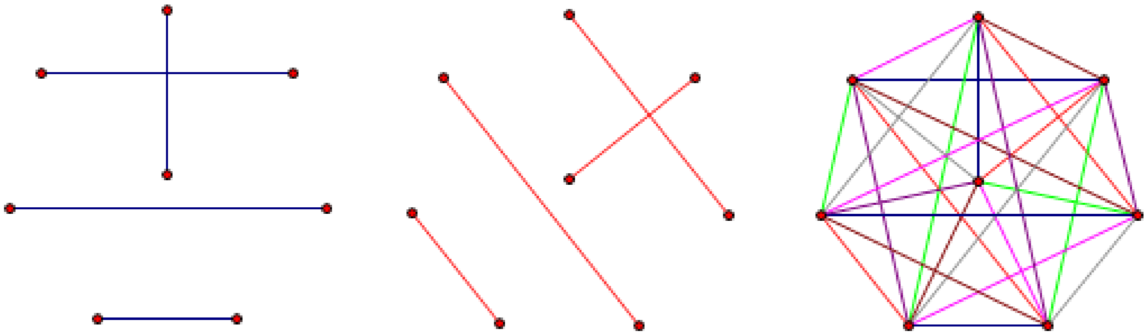


Figure 36: The 1-factorization GK_8

A 1-factorization of K_{2n} is said to be *rotational* if it is stabilized by a permutation of $[2n]$ that fixes one vertex and cyclically permutes the rest. Indeed, the 1-factorization GK_8 illustrated above is an example, where we let ρ denote the permutation that fixes the center vertex and cyclically permutes the other vertices. Notice that ρ also cyclically permutes the seven 1-factors of this 1-factorization, thereby stabilizing the set of them. Analogously, we see that GK_{2n} is rotational for any n.

Given any 1-factorization $\mathcal{F}$ of K_{2n}, we say that a subgraph G is *orthogonal* to $\mathcal{F}$ if each 1-factor of $\mathcal{F}$ shares at most one edge with G. For example, the star, in which the center vertex is adjacent to all of the outer vertices, is a spanning tree for K_8 that is orthogonal to the 1-factorization GK_8 described above. One way to visualize orthogonal spanning trees is to use colors on the edges — given a 1-factorization $\mathcal{F}$, we imagine that each perfect matching in $\mathcal{F}$ colors its edges a distinct color. An orthogonal spanning tree, then, is often said to be *rainbow-colored* by $\mathcal{F}$, since it must contain one edge of each color.

We will be considering many different subgraphs of the complete graph K_{2n}. We continue to draw such graphs in *rotational form*, meaning vertices 1 through $2n-1$ are spaced evenly around a circle in clockwise manner, and vertex $2n$ is placed at the center. An example of a subgraph drawn in rotational form appeared in Figure 36 earlier.

Suppose K_{2n} is drawn in rotational form. For any edge $\{a, b\}$, if $2n \notin \{a, b\}$, we define its *length* by

$$\text{length}\{a, b\} = \min\{|a-b|, 2n-1-|a-b|\},$$

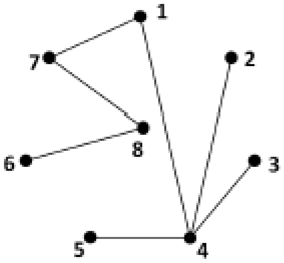
and if $2n \in \{a, b\}$, we say the edge has length 0.

For any edge $\{a, b\}$ of nonzero length, we define its *center* (denoted center$\{a, b\}$) to be the unique vertex $x \in [2n-1]$ such that

$$\text{length}\{a, x\} = \text{length}\{b, x\}.$$

An edge $\{2n, b\}$ of length 0 has center b.

Figure 37 gives an example of a subgraph G of K_8 drawn in rotational form, along with information about its edges.



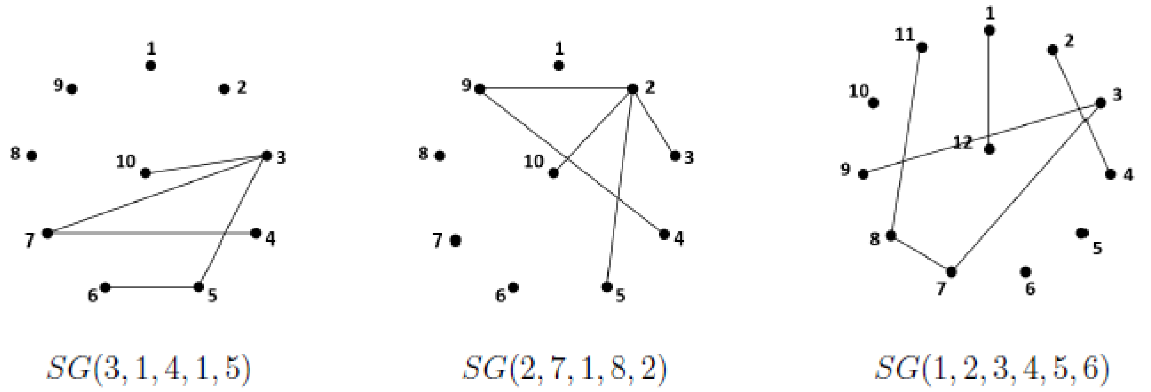
edge	1, 4	1, 7	2, 4	3, 4	4, 5	6, 8	7, 8
length	3	1	2	1	1	0	0
center	6	4	3	7	1	6	7

Figure 37: Lengths and centers of edges in a rotational drawing of K_8

Note that any edge in K_{2n} is uniquely determined by its center and length.

To obtain a rotational decomposition of K_{2n}, we begin with starter graphs, which are graphs that contain one edge of each possible length.

Fix any integer $n > 0$ and any n-tuple of integers $(c_0, \ldots, c_{n-1})$ satisfying $0 < c_l < 2n$ for each l $(0 \leq l < n)$. We define the *starter graph*, denoted $SG(c_0, \ldots, c_{n-1})$, to be the subgraph of K_{2n} with a single edge of length l and center c_l, for each l $(0 \leq l < n)$. Figure 38 illustrates examples of starter graphs.



$SG(3,1,4,1,5)$ $SG(2,7,1,8,2)$ $SG(1,2,3,4,5,6)$

Figure 38: Examples of starter graphs

Suppose G is a subgraph of K_{2n} with edge set E. Let ρ denote the permutation of $[2n]$ that cyclically permutes $(1,2,3,\ldots,2n-1)$ and fixes $2n$. We define $\rho(G)$ to be the subgraph of K_{2n} with edge set

$$\rho(E) = \{\{\rho(a),\rho(b)\} \mid \{a,b\} \in E\}.$$

Equivalently, we can simply say that vertices a, b are adjacent in $\rho(G)$ if and only if the vertices $\rho^{-1}(a)$, $\rho^{-1}(b)$ are adjacent in G.

Using the above, we can now introduce the notion of a rotational family of subgraphs.

Fix any integers $n, d > 0$ and let G be any subgraph of K_{2n}. We define the *rotational family* $\mathcal{F}_G^d$ generated by G to be the set

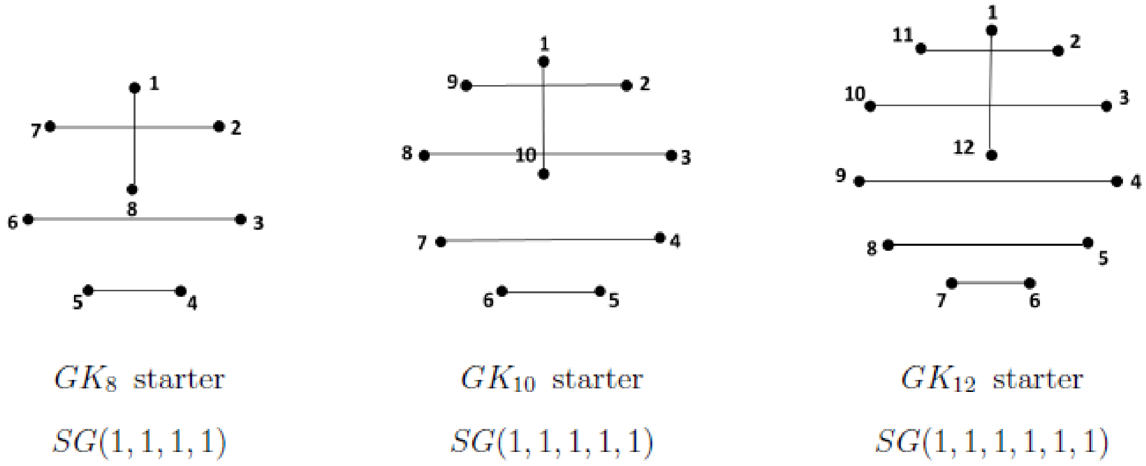
$$\mathcal{F}_G^d := \{G, \rho(G), \ldots, \rho^{d-1}(G)\}.$$

Notice that the set $\mathcal{F}_G^d$ has cardinality d.

We will illustrate our results using the following two families of rotational 1-factorizations, the first of which is fairly common in the literature.

Fix any integer $n > 0$. The 1-factorization GK_{2n} of K_{2n} is the rotational family $\mathcal{F}_G^{2n-1}$ generated by the graph $G = SG(c_0, \ldots, c_{n-1})$ where

$$c_l = 1 \text{ for all } l \ (0 \leq l < n).$$



GK_8 starter

$SG(1, 1, 1, 1)$

GK_{10} starter

$SG(1, 1, 1, 1, 1)$

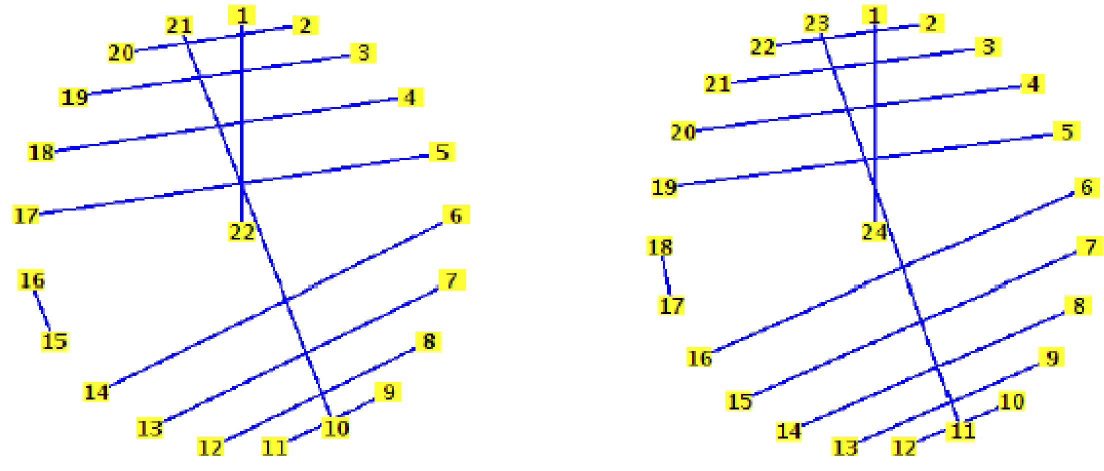
GK_{12} starter

$SG(1, 1, 1, 1, 1, 1)$

Figure 39: Starter graphs for the GK_{2n} 1-factorization

A new family of rotational 1-factors was discovered. The *half family*, HK_{2n}, is constructed by essentially splitting up the even and odd length edges and grouping them together. More formally, we have the following. Fix any integer $n > 0$. The 1-factorization HK_{2n} of K_{2n} is the rotational family $\mathcal{F}_G^{2n-1}$ generated by the graph $G = SG(c_0, \ldots, c_{n-1})$ where c_l is given by the following chart, depending on the form of n and l:

	$l < 2$		$2 \leq l \leq n-2$		$l > n-2$
	$l = 0$	$l = 1$	$l \equiv_2 0$	$l \equiv_2 1$	$l = n-1$
$n = 2k+0$	1	k	$2k-1$	$2k$	$3k-1$
$n = 2k+1$	1	k	$2k$	$2k+1$	k



HK_{22} starter

$SG(1,5,10,11,10,11,10,11,10,11,5)$

HK_{24} starter

$SG(1,6,11,12,11,12,11,12,11,12,11,17)$

Figure 40: Starter graphs for the HK_{2n} 1-factorization

In order to construct spanning trees that are orthogonal to a given rotational 1-factorization, we use rotational families of graphs that are built using opposing pairs of edges.

Suppose K_{2n} is drawn in rotational form and let e_1, e_2 be any pair of edges with centers c_1, c_2. We define the *distance* between them to be

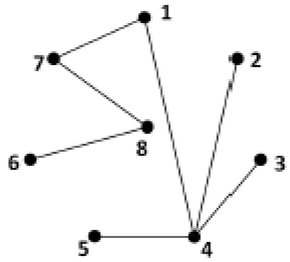
$$\text{dist}(e_1, e_2) = \text{length}\{c_1, c_2\},$$

where edges at distance $n-1$ are said to be *opposing*.

We next define the *direction* of the pair e_1, e_2 with centers c_1, c_2 to be the vertex

$$\text{dir}(e_1, e_2) = \text{center}\{c_1, c_2\}.$$

Figure 41 gives an example of a subgraph G of K_8 drawn in rotational form, along with information about its edge pairs.

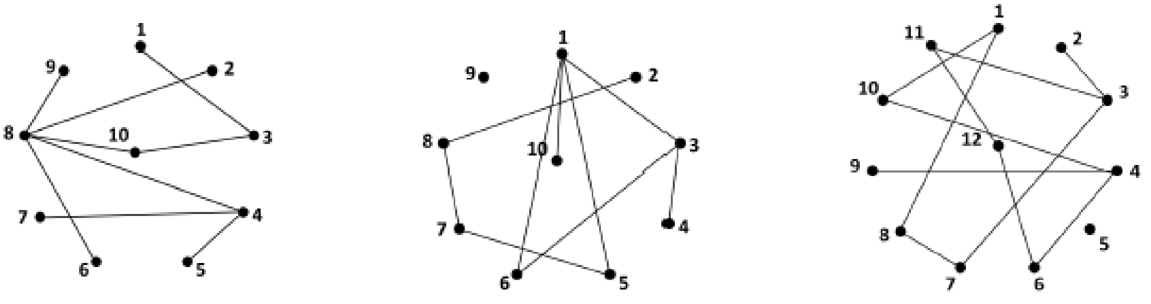


edge pair	$17, 34$	$17, 45$	$34, 45$	$68, 78$
distance	3	3	1	1
direction	2	6	4	3

Figure 41: Distance and direction for edge pairs in a rotational drawing of K_8

With this terminology in place, we can now define the graphs of interest.

Fix any integers $n > t \geq 0$ and any n-tuple of integers $(d_0, \ldots, d_{n-1})$ satisfying $0 < d_i < 2n$ for each i $(0 \leq i < n)$. We define the *opposing pair graph*, denoted $OPG_t(d_0, \ldots, d_{n-1})$, to be the subgraph of K_{2n} with a single edge of length t and center d_t, and, for each $i \neq t$, an opposing pair of edges of length i and direction d_i. We refer to t as the *exceptional length* of the OPG. Figure 42 gives examples of opposing pair graphs.



$OPG_4(1, 2, 9, 3, 6)$ $OPG_0(1, 1, 4, 7, 1)$ $OPG_3(3, 5, 8, 7, 2, 4)$

Figure 42: Examples of opposing pair graphs

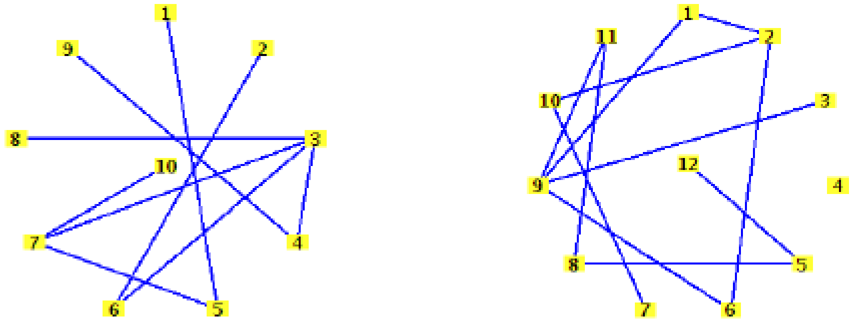
Notice that any opposing pair graph in K_{2n} has exactly $2n-1$ edges, which is the same as the number of edges required for a spanning tree of K_{2n}. Indeed, the graph $OPG_4(1, 2, 9, 3, 6)$, which is the left-most graph depicted above, is a spanning tree for

K_{10}. In general, as the center graph above shows, an opposing pair graph need not be acyclic. By a standard result about spanning trees [32, Theorem 2.1.4], if S is any set of $2n-1$ edges in K_{2n}, then S will be a spanning tree for K_{2n} if and only if S is acyclic and if and only if S forms a connected graph on the vertex set $[2n]$.

We define the *rest graph* of G by

$$\mathrm{Rest}(G) = K_{2n} - \left(\bigcup_{H \in \mathcal{F}_G^{n-1}} H \right).$$

In other words, $\mathrm{Rest}(G)$ is the complementary graph in K_{2n} of the union of the rotational family $\mathcal{F}_G^{n-1}$. Figure 43 illustrates the rest graphs for two of the previous opposing pair graphs.



$\mathrm{Rest}(OPG_4(1,2,9,3,6))$ $\mathrm{Rest}(OPG_3(3,5,8,7,2,4))$

Figure 43: Examples of rest graphs

It should be noted that when $G = OPG_t(d_0, \ldots, d_{n-1})$, the rest graph $\mathrm{Rest}(G)$ has exactly n edges of length t and a single edge of each length $i \neq t$, for a total of $2n-1$ edges. Occasionally it will happen that both G and $\mathrm{Rest}(G)$ are spanning trees for K_{2n}. When this occurs, the graph G will be of great use in advancing our goal of constructing an orthogonal spanning tree decomposition.

5.2 Literature

The first result on this topic goes back to work by Brualdi and Hollingsworth, who showed that any 1-factorization of the even complete graph admits a pair of disjoint orthogonal spanning trees.

Theorem 5.1. *[3, Theorem 1] Any 1-factorization of K_{2n} has at least two disjoint orthogonal spanning trees.*

Soon thereafter, this was extended to include a third tree.

Theorem 5.2. *[19] Any 1-factorization of K_{2n} has at least three disjoint orthogonal spanning trees.*

For the particular 1-factorization GK_{2n}, and for certain values of n, Krussel, et al. showed that it is possible to improve tremendously on the number of disjoint orthogonal spanning trees.

Theorem 5.3. *[19] If $2n-1$ is a prime of the form $8m+7$, there exists a full set of n disjoint orthogonal spanning trees for GK_{2n}.*

The above result uses number-theoretic properties in the construction of the orthogonal trees, but it did not, perhaps, take full advantage of the rotational symmetry of the 1-factorization GK_{2n}. This observation led to a new avenue of investigation described next.

5.3 New Results

Using the rotational symmetry of GK_{2n} and HK_{2n}, a strengthening of the results in the last section can be obtained.

Theorem 5.4. *[14, Theorem 7.1] For every positive integer n, there exists a full set of n disjoint orthogonal spanning trees for GK_{2n}.*

Using the techniques developed in the proof of the result above, there appears to be room for further extensions.

Theorem 5.5. *For any positive integer $3 \leq n \leq 11$, there exists a full set of n disjoint spanning trees for K_{2n} orthogonal to HK_{2n}.*

For the data that supports this result, see Table 4 in Appendix E.

We conjecture that for any positive integer n, there exists a full set of n disjoint spanning trees for K_{2n} that are orthogonal to HK_{2n}. We have generated a large amount of data that supports this conjecture. As n increases, the number of distinct opposing pair graphs that generate a decomposition orthogonal to HK_{2n} grows rapidly. See Appendix E for a summary of the data collected.

Given a starter graph 1-factor S, we would like to explore the set of opposing pair graphs that are orthogonal to S to see which ones are trees that lead to complete spanning tree decompositions of K_{2n}. Suppose t, the length of the exceptional edge, is fixed. For each of the n edge lengths, we have $2n-1$ choices for the direction of their corresponding pair (or single edge, in the case of the exceptional length edge). This seems to give us $(2n-1)^n$ opposing pair graphs to check. In fact, our search space is much smaller and all of the orthogonal OPGs can be easily enumerated, as the next results show.

Theorem 5.6. *[14, Theorem 6.2] Fix integers $n > t \geq 0$ and any two n-tuples of integers $(c_0, \ldots, c_{n-1})$ and $(d_0, \ldots, d_{n-1})$ satisfying $0 < c_i, d_i < 2n$ for each i $(0 \leq i < n)$. By rotation, assume $d_t = c_t$. Let $S = SG(c_0, \ldots, c_{n-1})$ and $G = OPG_t(d_0, \ldots, d_{n-1})$.*

Then the graph G is orthogonal to S if and only if

$$\{d_i - c_i + (-1)^n \lfloor n/2 \rfloor\}_{i \neq t} \equiv \{1, 2, \ldots, n-1\} \pmod{2n-1}.$$

Corollary 5.7. *[14, Corollary 6.3] Each rotational 1-factorization of K_{2n} and choice of $0 \leq t < n$ gives rise to $(n-1)!$ different orthogonal opposing pair graphs.*

The challenge is now to see which of those orthogonal graphs are also spanning trees. We can enumerate the $(n-1)!$ tuples corresponding to all of the OPGs orthogonal to the starter graph 1-factor and test to see which of them gives a tree decomposition of K_{2n}. The procedure goes as follows.

1. Choose the color c for the exceptional length edge t.

2. Find the *symmetric opposite sequence* (SOS) of c, the $n-1$ numbers symmetric about the line connecting c to vertex $2n$ in the rotational form of K_{2n}, read clockwise.

3. Now we build a table. Write out the tuple for the starter graph 1-factor, then create $2n-2$ more rows, with the value in each position deriving from the one above it by being the next value clockwise from it in the rotational form of K_{2n}.

4. Record the subtable that contains the SOS in the t^{th} column. The top rows might need to be cycled back to the bottom in order to find the complete SOS.

5. Remove the t^{th} column.

6. The directions for the edge pairs for the OPGs can now be found by performing a traversal of the subtable. Choose $n-1$ values such that exactly one value is chosen from each row and column. The respective OPG tuples are built by taking these values and inserting c into the t^{th} position.

This procedure is explored in the following example. Suppose our 1-factor is $S = SG(1, 2, 4, 5, 2)$ and $t = 4$. WOLOG let the color of the exceptional length edge be $c = 1$. This gives us the SOS of 4,5,6,7. Here is the table generated by S.

1	2	4	5	2
2	3	5	6	3
3	4	6	7	**4**
4	5	7	8	**5**
5	6	8	9	**6**
6	7	9	1	**7**
7	8	1	2	8
8	9	2	3	9
9	1	3	4	1

We see the SOS in bold in the last column corresponding to $t = 4$. We take the following subtable based on the location of the SOS.

3	4	6	7
4	5	7	8
5	6	8	9
6	7	9	1

Any traversal of this subtable will give the values $d_0, \ldots, d_3$ that, when combined with $d_t = d_4 = c = 1$, will give an orthogonal opposing pair graph. For example, $OPG_4(3, 5, 8, 1, 1)$ will be orthogonal to S.

5.4 Conjectures

In there original paper, Brualdi and Hollingsworth made an extremely strong conjecture that remains very open.

Conjecture 5.8. *[3, Conjecture 1] For any 1-factorization of K_{2n}, there exists a full set of n disjoint orthogonal spanning trees.*

Figure 44 shows a 1-factorization of K_6 along with a full set of three disjoint orthogonal spanning trees.

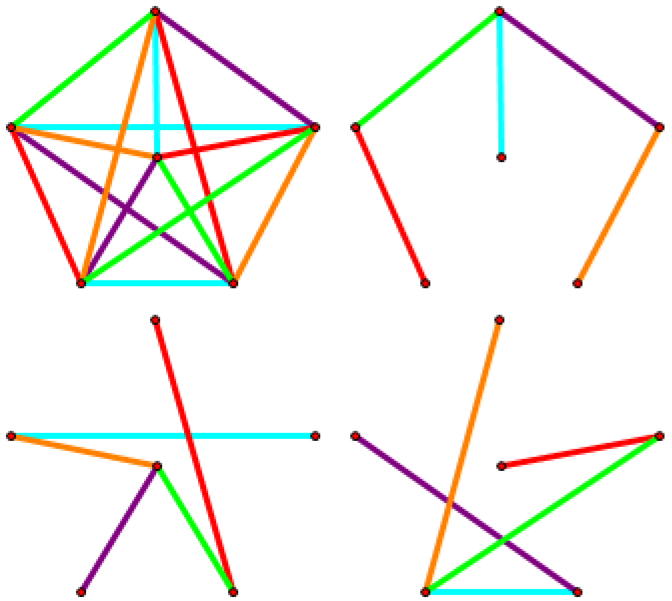


Figure 44: A successful test of the conjecture for K_6

We put forth several restricted versions of this conjecture that we hope will be easier to attack.

Conjecture 5.9. *For any positive integer n, there exists a full set of n disjoint spanning trees for K_{2n} that are orthogonal to HK_{2n}.*

Conjecture 5.10. *[14] Every rotational 1-factorization of K_{2n} has a full set of n disjoint rotational orthogonal spanning trees.*

In future work, we propose to analyze the data we have, which supports these conjectures, to extend the earlier work and obtain stronger results. The new HK_{2n} family is a good candidate for a rotational 1-factor with easily describable orthogonal spanning trees. See Appendix E for data concerning this family.

References

[1] B. Alspach and G. Liu. Paths and cycles in matroid basis graphs. *Graphs and Combinatorics*, 5(1):207–211, 1989.

[2] J. A. Bondy and A. W. Ingleton. Pancyclic graphs ii. *Journal of Combinatorial Theory, Series B*, 20(1):41–46, 1976.

[3] R. A. Brualdi and S. Hollingsworth. Multicolored trees in complete graphs. *Journal of Combinatorial Theory*, 68(2):310–313, 1996.

[4] R. L. Cummins. Hamilton circuits in tree graphs. *Circuit Theory, IEEE Transactions on*, 13(1):82–90, 1966.

[5] D. M. Cvetković, M. Doob and H. Sachs. *Spectra of graphs: theory and application*, volume 87. Academic Pr, 1980.

[6] E. S. El-Mallah and C. J. Colbourn. The complexity of some edge deletion problems. *Circuits and Systems, IEEE Transactions on*, 35(3):354–362, 1988.

[7] R. C. Entringer and P. J. Slater. On the maximum number of cycles in a graph. *Ars Combin*, 11:289–294, 1981.

[8] R. Frucht. Herstellung von graphen mit vorgegebener abstrakter gruppe. *Compositio Mathematica*, 6:239–250, 1939.

[9] C. Godsil and G. Royle. *Algebraic Graph Theory.* Springer, 2001.

[10] C. A. Holzmann and F. Harary. On the tree graph of a matroid. *SIAM Journal on Applied Mathematics*, 22(2):187–193, 1972.

[11] T. Kamae. The existence of a hamiltonian circuit in a tree graph. *IEEE Transactions on Circuit Theory*, 14(3):279–283, 1967.

[12] J. Krausz. Démonstration nouvelle d'une théorème de Whitney sur les réseaux. *Mat. Fiz. Lapok*, 50:75–85, 1943. Hungarian.

[13] M. S. Krishnamoorthy and N. Deo. Tree graphs and tree numbers. *Circuits and Systems, IEEE Transactions on*, 22(1):60–62, 1975.

[14] J. Caughman, J. Krussel and J. Mahoney. Spanning tree decompositions of K_{2n} orthogonal to rotational 1-factorizations. *Submitted.*

[15] J. Lauri and R. Scapellato. *Topics in graph automorphisms and reconstruction.* Cambridge University Press, 2003.

[16] G. Liu. Paths between two vertices in a tree graph. *Systems Science and Mathematical Sciences*, 5(2):108–110, 1992.

[17] H. Liu. Bounds of edge chromatic number of tree graph. *Journal of Mathematics*, 2002(1), 2002. In Chinese.

[18] L. Lovász. *Combinatorial Problems and Exercises.* Akadémiai Kiadó, Budapest, 1979.

[19] J. Krussel, S. Marshall and H. Verral. Spanning trees orthogonal to one-factorizations of K_{2n}. *Ars. Combin.*, 57:77–82, 2000.

[20] S. B. Maurer. Matroid basis graphs. i. *Journal of Combinatorial Theory, Series B*, 14(3):216–240, 1973.

[21] S. B. Maurer. Matroid basis graphs. ii. *Journal of Combinatorial Theory, Series B*, 15(2):121–145, 1973.

[22] B. D. McKay and A. Piperno. Practical graph isomorphism, {II}. *Journal of Symbolic Computation*, 60(0):94 – 112, 2014.

[23] K. Menger. Zur allgemeinen kurventheorie. *Fundamenta Mathematicae*, 10(1):96–115, 1927.

[24] M. C. Hernando, F. Hurtado, A. Márquez, M. Mora and M. Noy. Geometric tree graphs of points in convex position. *Discrete Applied Mathematics*, 93(1):51–66, 1999.

[25] X. Li, V. Neumann-Lara and E. Rivera-Campo. On a tree graph defined by a set of cycles. *Discrete mathematics*, 277(1):303–310, 2003.

[26] V. Estivill-Castro, M. Noy and J. Urrutia. On the chromatic number of tree graphs. *Discrete Mathematics*, 223(1):363–366, 2000.

[27] G. Sabidussi. Graphs with given group and given graph-theoretical properties. *Canadian Journal of Mathematics*, 9:515–525, 1957.

[28] Z. Stanić. Some notes on minimal self-centered graphs. *AKCE J. Graphs. Combin.*, 7(1):97–102, 2010.

[29] V. G. Vizing. On an estimate of the chromatic class of a p-graph. *Diskret. Analiz*, 3(7):25–30, 1964.

[30] K. Wagner. Über eine eigenschaft der ebenen komplexe. *Mathematische Annalen*, 114(1):570–590, 1937.

[31] D. J. A. Welsh. *Matroid Theory.* Academic Press, 1976.

[32] D. B. West. *Introduction to Graph Theory*. Pearson Education, Inc., 2nd edition, 2001.

[33] H. Whitney. Congruent graphs and the connectivity of graphs. *American Journal of Mathematics (The Johns Hopkins University Press)*, 54(1):150–168, 1932.

[34] H. Whitney. 2-isomorphic graphs. *American Journal of Mathematics*, pages 245–254, 1933.

Appendices

Appendix A Selected Algorithms

To carry out the original research for this paper, a program was written that takes a graph as input and outputs its tree graph. Here is an outline of the algorithm.

1. Input the base graph G
2. Find $\tau(G)$, the number of spanning trees of G using the Matrix Tree Theorem
3. Find and list the edges in $E(G)$
4. Create C, the set of every possible $(n-1)$-set of edges from $E(G)$
5. For each set of edges in C, see if it forms a spanning tree of G
    - If so, record it and increase the tally of found trees
6. Stop once the tally reaches $\tau(G)$
7. For every pair of stored spanning trees, see if they have the edge exchange property
    - If so, put an edge between them in $\mathcal{T}(G)$
8. Output $\mathcal{T}(G)$

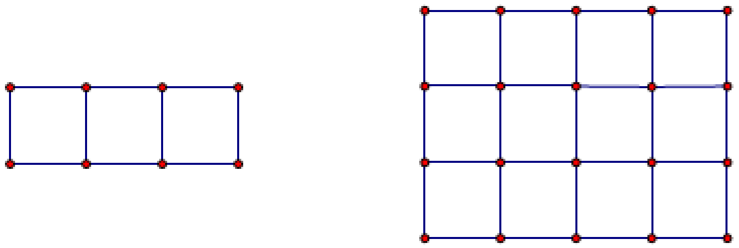
Additional programs were written to calculate various graph parameters of tree graphs, such as the vertex-connectivity and center. In addition, the free *nauty* [22] software was used extensively to investigate the automorphism groups of tree graphs.

Appendix B Examples of Graph Families

This appendix illustrates some of the named families of graphs mentioned in the paper.

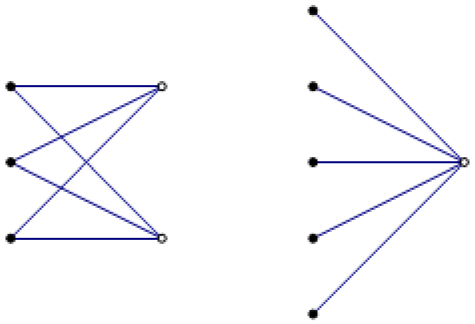
B.1 Grid Graphs

$\text{Grid}_{4,2}$ and $\text{Grid}_{5,4}$

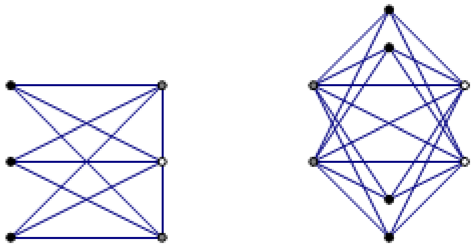


B.2 Complete Multipartite Graphs

$K_{3,2}$ and $K_{5,1}$

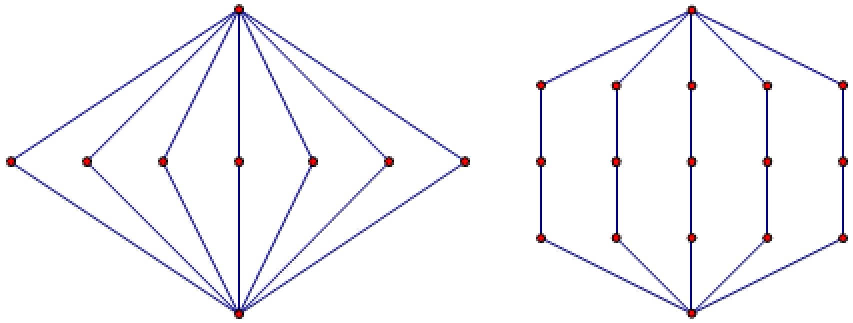


$K_{3,2,1}$ and $K_{4,2,2}$



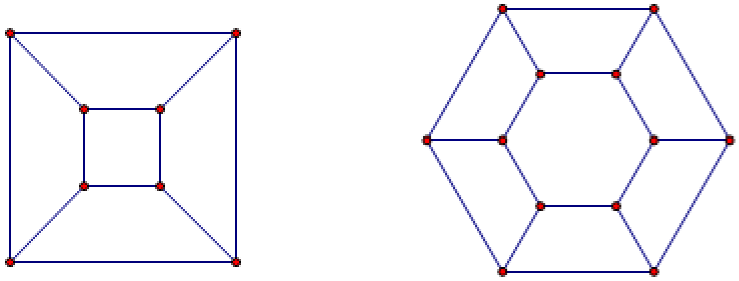
B.3 $P_{n,k}$ Graphs

$P_{7,2}$ and $P_{5,4}$



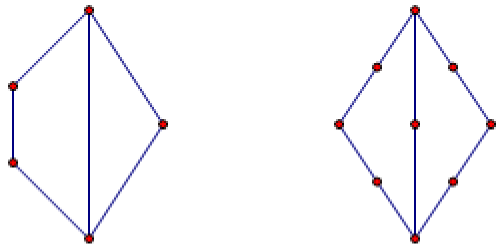
B.4 Prism Graphs

Prism_4 and Prism_6



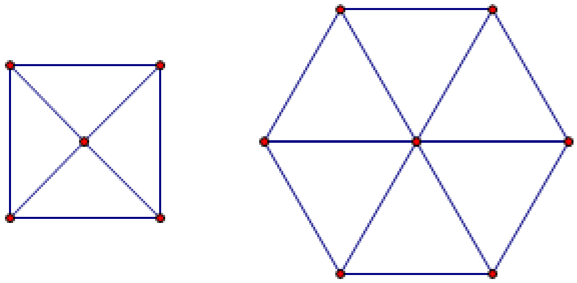
B.5 $\theta_{a,b,c}$ Graphs

$\theta_{3,2,1}$ and $\theta_{4,4,2}$



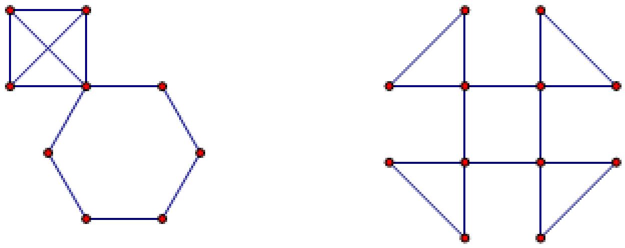
B.6 Wheel Graphs

W_5 and W_7



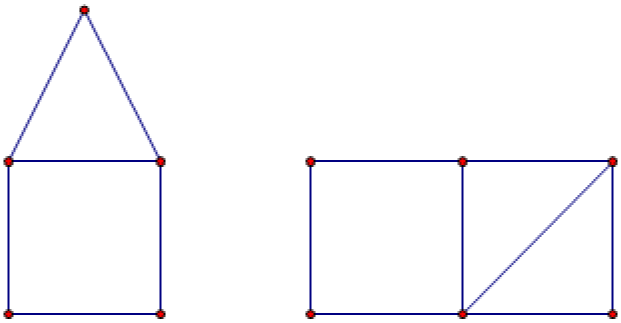
B.7 $\bigodot$ Graphs

$K_4 \bigodot C_6$ and $C_4 \bigodot C_3 \bigodot C_3 \bigodot C_3 \bigodot C_3$



B.8 Named Graphs

House and Asym6



Appendix C Spectrum Data from $\mathcal{T}(P_{n,2})$

The $\mathcal{T}(P_{n,k})$ graphs are conjectured (see Conjecture 4.35) to be integral with the following spectrum: There are $\lceil \frac{3n-2}{2} \rceil$ distinct eigenvalues, with maximum eigenvalue $(n-1)(2k-1)$. There are $\lceil \frac{n-1}{2} \rceil$ jumps of size $2k$ between the next biggest distinct eigenvalues, then jumps of size k down to the smallest eigenvalue of $1-n$. Table 1 shows the eigenvalues of some $\mathcal{T}(P_{n,2})$ graphs.

Table 1: Eigenvalues of $\mathcal{T}(P_{n,2})$

Graph G	Eigenvalues of $\mathcal{T}(G)$													
$K_{2,2}$	-1	3												
$K_{3,2}$	-2	0	2	6										
$K_{4,2}$	-3	-1	1	5	9									
$K_{5,2}$	-4	-2	0	2	4	8	12							
$K_{6,2}$	-5	-3	-1	1	3	7	11	15						
$K_{7,2}$	-6	-4	-2	0	2	4	6	10	14	18				
$K_{8,2}$	-7	-5	-3	-1	1	3	5	9	13	17	21			
$K_{9,2}$	-8	-6	-4	-2	0	2	4	6	8	12	16	20	24	
$K_{10,2}$	-9	-7	-5	-3	-1	1	3	5	7	11	15	19	23	27

Patterns were also found in the multiplicities of eigenvalues of $\mathcal{T}(P_{n,2})$.

Table 2: Multiplicities of eigenvalues of $\mathcal{T}(P_{n,2})$

3	1												
6	2	3	1										
10	8	9	4	1									
15	20	25	4	10	5	1							
21	40	60	24	25	15	6	1						
28	70	126	84	70	6	35	21	7	1				
36	112	238	224	196	48	77	56	28	8	1			
45	168	414	504	504	216	189	8	126	84	36	9	1	
55	240	675	1008	1170	720	525	80	261	210	120	45	10	1

The i^{th} row of Table 2 shows the multiplicities of the eigenvalues of $\mathcal{T}(P_{i+1,2})$, in order from smallest to largest eigenvalue. The patterns appear as we read the diagonals from the array. For example, the leftmost values, $3, 6, 10, \ldots$, are the triangular numbers. The formula for these is $\binom{n+1}{2}$. The second-from-the-left values, $2, 8, 20, \ldots$, come from $2\binom{n}{3}$. Fourth-from-the-left starting with $4, 24, 84, \ldots$, come from $4\binom{n}{5}$. Sixth-from-the-left starting with $6, 48, 216, \ldots$, come from $6\binom{n}{7}$. In general, the pattern for these even diagonals d from the left seems to be $d\binom{n}{d+1}$. The pattern for the third-from-the-left, $9, 25, 60, \ldots$, seems to be $(n^3 - 5n^2 + 10n)/8$; no binomial coefficient here. The rest of the odd diagonals do not seem to want to give up a pattern, despite many visits to the Online Encyclopedia of Integer Sequences.

Coming from the right side now, the patterns are more straightforward. $1, 1, 1, \ldots$ is $\binom{n}{0}$. $3, 4, 5, \ldots$ is $\binom{n}{1}$. $10, 15, 21, \ldots$ is $\binom{n}{2}$. The pattern for the d^{th} diagonal from the right seems to be $\binom{n}{d-1}$. No explanation for the behavior of these multiplicities is immediately apparent. Stare too long at the patterns and the patterns stare back.

Appendix D Catalog of Tree Graph Data

Table 3 is a collection of some of the data generated during this research on tree graphs. The first column is the input graph G, and the remaining columns refer to parameters of its tree graph $\mathcal{T}(G)$. The programs used to calculate certain graph parameters, such as the independence number (α), the clique number (ω), and the chromatic number (χ), did not run well on graphs larger than about 50 vertices, and so many of those entries are blank in the table.

This second half contains parameters from the base graphs. The μ parameter can be difficult to measure for larger, more complicated graphs, so is missing from some of these rows.

Table 3: Portion of tree graph data collected

	Parameters for G							Parameters for $\mathcal{T}(G)$								
Graph G	n	m	v	$girth$	$circ$	μ	$\|Aut\|$	n	m	α	ω	δ	Δ	χ	$diam$	$\|Aut\|$
$K_{1,1,1}$	3	3	1	3	3	1	6	3	3	1	3	2	2	3	1	6
$K_{2,1,1}$	4	5	2	3	4	5	4	8	18	2	4	4	5	4	2	8
$K_{3,1,1}$	5	7	3	3	4	18	12	20	72	5	4	6	8	4	3	48
$K_{4,1,1}$	6	9	4	3	4	56	48	48	240	10	5	8	11	6	4	384
$K_{5,1,1}$	7	11	5	3	4	160	240	112	720			10	14		5	3840
$K_{6,1,1}$	8	13	6	3	4	432	1440	256	2016			12	17		6	46080
$K_{3,2,1}$	6	11	6	3	6		12	216	1800			12	20		5	12
$K_{2,2,2}$	6	12	7	3	6		48	384	3768			17	23		5	48
$K_{3,2}$	5	6	2	4	4	6	12	12	36	3	4	6	6	4	2	48
$K_{4,2}$	6	8	3	4	4	24	48	32	144	8	4	9	9	4	3	384
$K_{5,2}$	7	10	4	4	4	80	240	80	480			12	12		4	3840
$K_{6,2}$	8	12	5	4	4	240	1440	192	1440			15	15		5	46080
$K_{7,2}$	9	14	6	4	4	672	10080	448	4032			18	18		6	645120
$K_{3,3}$	6	9	4	4	6	39	72	81	522			12	14		4	72
$K_{4,3}$	7	12	6	4	6		144	432	4320			18	22		6	144
$K_{5,3}$	8	15	8	4	6		720	2025	27540			24	30		7	720
K_4	4	6	3	3	4	15	24	16	54	4	4	6	7	5	3	48
K_5	5	10	6	3	5		120	125	930			12	16		4	120
K_6	6	15	10	3	6		720	1296	17460			20	30		5	720
K_7	7	21	15	3	7		5040	16807	365085			30	50			5040
$\text{Grid}_{3,2}$	6	7	2	4	6	7	4	15	51	3	6	6	8	6	2	72
$\text{Grid}_{3,3}$	9	12	4	4	8		8	192	1536			12	22		4	128
$\text{Grid}_{4,2}$	8	10	3	4	8		4	56	304			9	15		3	144
$\text{Grid}_{4,3}$	12	17	6	4	12		4	2415	31677			18	44		6	64
$\text{Grid}_{5,2}$	10	13	4	4	10		4	209	1566			12	24		4	288

	Parameters for G							Parameters for $\mathcal{T}(G)$								
Graph G	n	m	v	$girth$	$circ$	μ	$\|Aut\|$	n	m	α	ω	δ	Δ	χ	$diam$	$\|Aut\|$
House	5	6	2	3	5	6	2	11	31	3	5	5	7	5	2	12
Asym6	6	8	3	3	6	24	1	29	122	7	6	7	12	6	3	12
$\theta_{4,2,1}$	6	7	2	3	6	7	2	14	47	3	6	6	9	6	2	48
$\theta_{5,2,1}$	7	8	2	3	7	8	2	17	66	3	7	7	11	7	2	240
$\theta_{4,3,1}$	7	8	2	4	7	8	2	19	75	4	7	7	10	7	2	144
$\theta_{5,3,1}$	8	9	2	4	8	9	2	23	103	4	8	8	12	8	2	720
$\theta_{4,4,1}$	8	9	2	5	8	9	4	24	108	4	8	8	11	8	2	1152
$\theta_{5,4,1}$	9	10	2	6	9	10	2	29	146	5	9	9	13	9	2	2880
$\theta_{3,2,2}$	6	7	2	4	5	7	4	16	58	3	5	7	8	6	2	48
$\theta_{3,3,2}$	7	8	2	5	6	8	8	21	90	4	6	8	9	6	2	144
W_5	5	8	4	3	5	32	8	45	232	11	5	8	12	5	4	16
W_6	6	10	5	3	6		10	121	830			10	18		5	20
W_7	7	12	6	3	7		12	320	2712			12	25		6	24
W_8	8	14	7	3	8		14	841	8428			14	33		7	28
Prism$_3$	6	9	4	3	6		12	75	465			10	14		4	12
Prism$_4$	8	12	5	4	8		48	384	3768			17	23		5	48
Prism$_5$	10	15	6	4	10		20	1805	24390			22	34		6	20
$P_{3,3}$	8	9	2	6	6	9	12	27	135	4	6	10	10		2	1296
$P_{3,4}$	11	12	2	8	8	12	12	48	336	6	8	14	14		2	82944
$P_{3,5}$	14	15	2	10	10	15	12	75	675			18	18		2	10368000
$P_{4,3}$	10	12	3	6	6	18	48	108	810			15	15		3	31104
$P_{4,4}$	14	16	3	8	8	24	48	256	2688			21	21		3	7962624
$P_{5,3}$	12	15	4	6	6	30	240	405	4050			20	20		4	933120
$P_{5,4}$	17	20	4	8	8	40	240	1280	17920			28	28		4	955514880

Appendix E Half Family Data

Table 4 gives opposing pair graphs that lead to a decomposition of K_{2n} orthogonal to the 1-factorization HK_{2n} for n from 3 to 11, supporting Theorem 5.5.

Table 4: Opposing pair graphs orthogonal to HK_{2n}

n	OPG orthogonal to HK_{2n}
3	$OPG_2(3, 4, 1)$
4	$OPG_3(4, 6, 5, 1)$
5	$OPG_4(5, 4, 7, 1, 1)$
6	$OPG_5(3, 9, 10, 9, 9, 1)$
7	$OPG_6(3, 10, 12, 10, 10, 12, 1)$
8	$OPG_7(3, 12, 14, 11, 13, 13, 11, 1)$
9	$OPG_8(3, 13, 16, 16, 11, 13, 13, 15, 1)$
10	$OPG_9(3, 15, 18, 18, 13, 16, 16, 15, 12, 1)$
11	$OPG_{10}(3, 16, 20, 15, 19, 17, 15, 19, 17, 14, 1)$

Table 5 gives the count of the number of distinct (up to rotation and reflection) opposing pair graphs orthogonal to the HK_{2n} 1-factorization for different values of n and exceptional lengths t. The counts for smaller values of t were not calculated for $n = 11$. This data suggests that Conjecture 5.9 is true, since as n increases there are many more available orthogonal opposing pair graphs with which to decompose K_{2n}.

Table 5: Number of different ways to decompose K_{2n} orthogonal to HK_{2n}

$n\backslash t$	0	1	2	3	4	5	6	7	8	9	10	Total
3	0	0	1									1
4	0	0	1	2								3
5	0	0	0	1	3							4
6	0	0	1	1	3	5						10
7	2	0	2	2	2	10	20					38
8	0	0	1	1	6	6	26	52				92
9	5	7	2	4	17	9	37	60	246			387
10	0	17	8	4	31	49	117	123	237	1074		1660
11											5251	≥ 5251