The Fixing Number of a Graph

A Major Qualifying Project Report

submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

by

Kara B. Greenfield

Date: 4/28/2011

Approved: 4/28/2011

Professor Peter R. Christopher, Advisor

Abstract

The fixing number of a graph is the order of the smallest subset of its vertex set such that assigning distinct labels to all of the vertices in that subset results in the trivial automorphism; this is a recently introduced parameter that provides a measure of the non-rigidity of a graph. We provide a survey of elementary results about fixing numbers. We examine known algorithms for computing the fixing numbers of graphs in general and algorithms which are applied only to trees. We also present and prove the correctness of new algorithms for both of those cases. We examine the distribution of fixing numbers of various classifications of graphs.

Executive Summary

This project began as an expansion on the work by Gibbons and Laison in [1]. In that paper, they defined the fixing number of a graph as the minimum number of vertices necessary to label so as to remove all automorphisms from that graph. A variety of open problems were proposed in that paper and the initial goals of this project were to solve as many of them as possible.

Gibbons and Laison proposed a greedy fixing algorithm for the computation of the fixing number of a graph and the first open question of their paper was whether the output of the algorithm was well defined for a given graph and further if that output would always equal the actual fixing number. We provide a counterexample to this algorithm to show that its output is not well defined for all possible input graphs. Additionally, we provide and prove the correctness of a new algorithm for the computation of fixing number of graphs in general. In addition to this algorithm we examine several existing algorithms for the computation of fixing numbers of trees and provide an enhancement of one of those algorithms that decreases its computational complexity.

In the final section of this paper, we examine the distribution of fixing numbers of several classes of graphs, i.e. determining the probability distribution of fixing numbers of graphs of a given order. We pay special attention to the fixing numbers of trees and of various sub-classifications of trees.

Acknowledgements

First and foremost, I would like to thank my advisor Professor Peter Christopher. As my academic advisor, he helped steer me towards my love of graph theory and as my MQP advisor, he proved to be an invaluable resource as both a collaborator and mentor. I would also like to thank Deborah Boutin for her assistance in preliminary research and proposing potential problems to me during the early stages of this project. Additionally, I would like to thank Professors Bill Martin and Mayer Humi for sharing their expert knowledge in some of the areas of this project with which I was previously unfamiliar.

Table of Contents

Table of Figures

Table of Algorithms

Chapter 1:
Basic Properties of Fixing Numbers

1 The Fixing Number

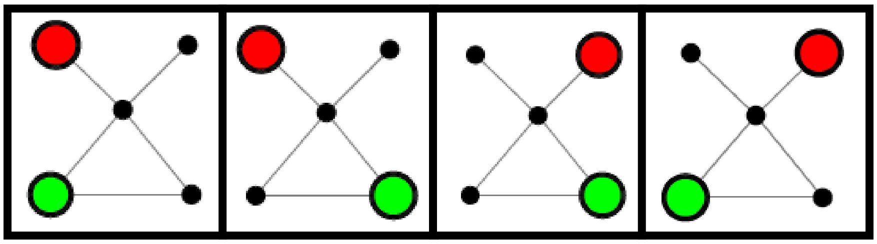


Figure 1 All Minimum Fixing Sets of a Graph

The fixing number was first defined by Erwin and Harary in 2006 in [2]. A fixing set is a set of vertices such that assigning a unique label to each vertex in that set removes all but the trivial automorphism from the graph. The fixing number of the graph is the order of the smallest fixing set. As shown in Figure 1, a given graph may have more than one fixing set of smallest possible order. This is permissible, as the fixing number is only concerned with the order of a minimum fixing set.

It is important to note that we are looking at minimum, not just minimal, sized fixing sets. For example, both sets highlighted vertices in the following graph form *minimal* fixing sets in that they fix the graph and the removal of any of the vertices from the set would render the graph not fixed. However, only the first fixing set is a *minimum* in that it contains the smallest possible number of vertices that still suffices to fix the graph.

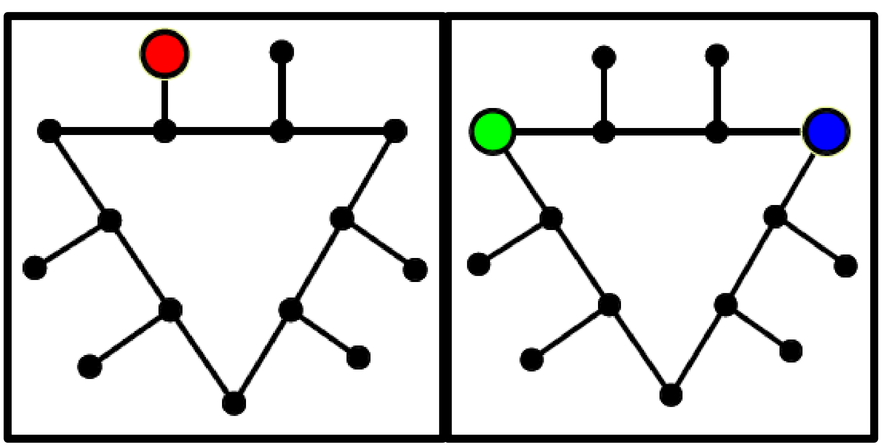


Figure 2 Minimal Fixing Sets

1.1 History of Fixing Numbers

The concept of a fixing number was first published, though not through a peer-reviewed medium by Josh Laison and Courtney Gibbons in “Fixing Numbers of Graphs and Groups” [1] on April 14th, 2006. The first official publication in a peer reviewed journal came exactly 4 months later when Frank Harary and David Erwin published in the Electronic Journal of Combinatorics [2]. In addition to having had the first peer reviewed publication on the topic, Erwin and Harary had submitted their work almost 2 years before Gibbons and Laison published their preprint. There was personal communication and preprints were shared between the two pairs of authors, so their naming conventions are consistent in that both use the term fixing number.

Figure 3 Pioneers in Fixing Numbers (from left to right: Frank Harary, Josh Laison, Debra Boutin)

Debra Boutin independently did research on fixing numbers under the name of “determining number” [3]. Boutin submitted her work to the Electronic Journal of Combinatorics shortly after Gibbons and Laison released their preprint, but before any work on the subject had been officially published in a peer reviewed journal. Consequently, she had done most of her work, and named the concept that she had independently defined, before having knowledge of the existence of other researchers working on the problem. She did have communication with the other authors before her work was published and too the opportunity to make her paper, “Identifying Graph Automorphisms Using Determining Sets” the first to acknowledge that there were two names for an equivalent graph property, but did nothing to resolve the issue of which should be used.

Currently Debra Boutin and Cáceres et.al. are the only groups who are actively doing research on the fixing number, with both using the terminology "determining number." As such, it is likely that this will become the more prominent term, but for the time being both determining number and fixing number are accepted in literature. We will use fixing number throughout this paper for purposes of historical accuracy.

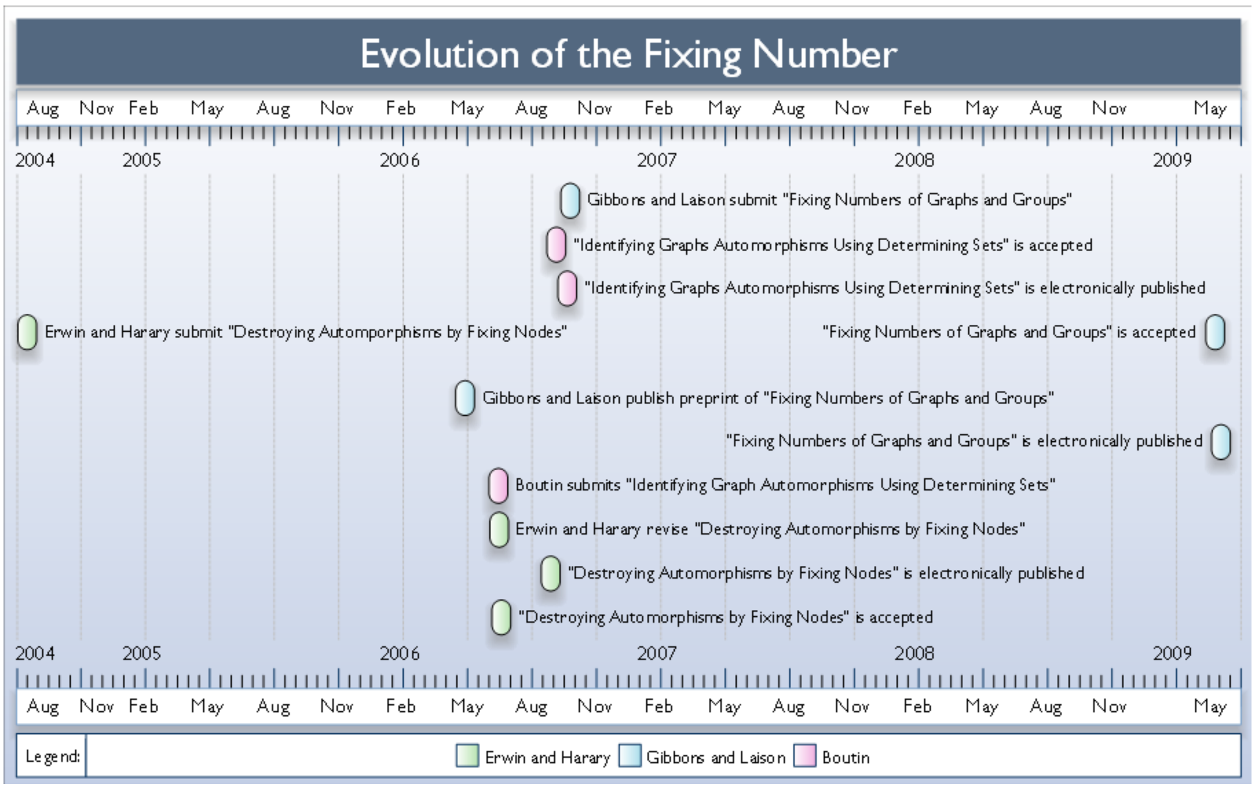


Figure 4 Early Publication History

2 Fixing Numbers of Special Graphs

The fixing numbers of some special classes of graphs are shown and proved below. These results are a combination of existing and original proofs. The fixing numbers of more special graphs are shown in Fixing Numbers of Special Graphs.

2.1 Fixing Number of the Complement of a Graph

Let G be a graph such that $G = (V, E)$. The complement of G is defined as

$$\bar{G} = (V, E' = \{(v_1, v_2) | v_1, v_2 \in V \text{ and } (v_1, v_2) \notin E\}) \qquad \textbf{Equation 1}$$

[4]

Theorem 1.

The fixing number of a graph is equal to the fixing number of its complement.

Proof:

Let $|V(G)| = n$. Two-color the edges of K_n so that one color corresponds to the graph G and the other color is used for the remaining edges. The 2-colored K_n and G have the same fixing number. The 2-colored K_n must have the same fixing number regardless of which color is used to define adjacency. Therefore,

$$\mathrm{fix}(G) = \mathrm{fix}(\bar{G}) \qquad \textbf{Equation 2}$$

■

This result is illustrated in the figure below.

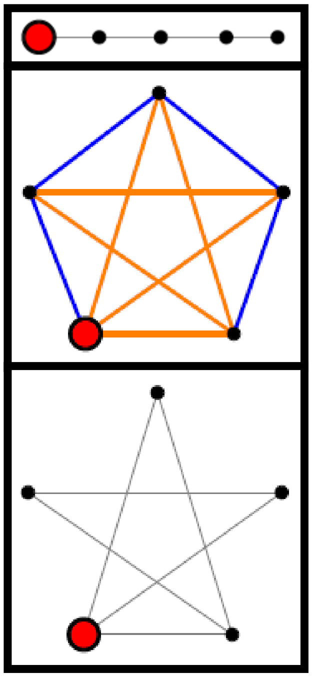


Figure 5 Fixing Number of a Complement

The fixing Number of P_5 is 1.

Two-color K_5 so that the blue edges correspond to P_5 and all other edges are orange. The fixing number of this graph is also 1.

Therefore, the fixing number of $\overline{P_5}$ is also equal to 1.

2.2 Fixing Number of Disconnected Graphs

A disconnected graph is a graph in which there exist 2 vertices with no path between them [5] [4]. A connected component of a graph is a maximal set of vertices and their adjoining edges such that there exists a path between every pair of vertices in the set.

Theorem 2.

Let G be a graph with $k > 1$ connected components, $H_1, \ldots, H_k$. We define a component class of G to be a maximal set of isomorphic connected components. Consider the component classes of G, which consist of isomorphic rigid components, where a rigid component is a component which if taken as a graph on its own has fixing number equal to 0 [6]. Note that the term rigid graph has an additional meaning to the one which we are using in this paper. Rigid graphs as we have defined them are alternatively known as asymmetric graphs or identity graphs [7]. Suppose there are j such rigid component classes with c_i components in the i^{th} component class. In other words, there will be j components that are mutually non-isomorphic and there will be c_i components which are isomorphic to the i^{th} component in the set of mutually non-isomorphic components.

$$fix(G) = \sum_{i=1}^{k} fix(H_i) + \sum_{i=1}^{j} (c_i - 1)$$

Equation 3

[2]

Proof:

Each connected component must be fixed and will require a fixing set of order $fix(H_i)$; additionally, the vertices that make up the fixing set for one connected component will necessarily be distinct from the those that make up the fixing set of any other connected component. Additionally, if two of the rigid components are isomorphic to each other, then one of them must have one vertex fixed in order to prevent the interchange of entire components.

■

As an example, consider the graph to the right, which consists of 6 connected components in 3 different component classes, 2 of which are rigid component classes. Therefore,

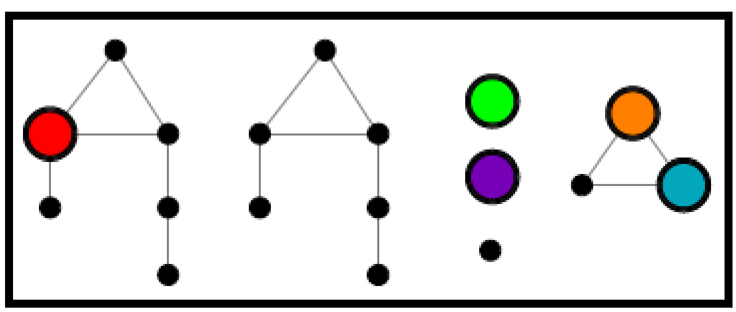
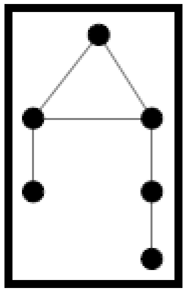
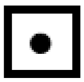
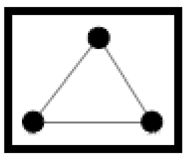


Figure 6 Fixed Disconnected Graph

- $k = 6$
- $j = 2$

i	Component Class H_i	Component Fixing Number $fix(H_i)$	Number of Components in Class c_i
1		0	2
2		0	3
3		1	1

$$fix(G) = \sum_{i=1}^{k} fix(H_i) + \sum_{i=1}^{j} (c_i - 1)$$

$$fix(G) = \sum_{i=1}^{6} fix(H_i) + \sum_{i=1}^{2} (c_i - 1)$$

$$fix(G) = (0 + 0 + 0 + 0 + 0 + 2) + \big((2 - 1) + (3 - 1)\big) = 2 + 1 + 2 = 5$$

With this understanding of graphs with multiple connected components, we will restrict the rest of our study to connected graphs.

2.3 Fixing Numbers of Cycles

The cycle graph C_n is a connected graph on n vertices which consists of a single cycle [8].

Theorem 3.

$$\forall n \geq 3, \mathrm{fix}(C_n) = 2$$ **Equation 4**

Proof:

The symmetries of a cycle are rotation and reflection. In a cycle, all of the vertices are in the same orbit. Without loss of generality, select one of the vertices to fix; call this vertex v. This removed rotational symmetry, but will not fix the graph, as reflection through the axis containing v will still be permissible. Therefore, the fixing number is greater than one. Now fix one of the vertices adjacent to v. This removes the remaining reflexive symmetry and thus fixes the graph. Therefore, the fixing number is 2. More generally, any 2 non-antipodal vertices of a cycle will form a fixing set [2].

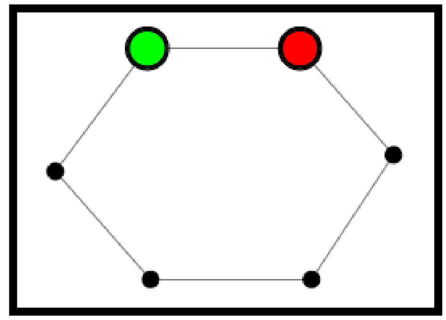


Figure 7 Cycle with an Example Minimum Fixing Set

■

2.4 Fixing Numbers of Paths

The path graph, P_n, is a tree of order n with two vertices of degree 1 and all other vertices of degree 2 [9].

Theorem 4.

$$\mathrm{fix}(P_1) = 0$$ **Equation 5**

$$\forall n > 1,\ \mathrm{fix}(P_n) = 1$$ **Equation 6**

Proof:

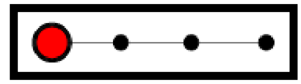


Figure 8 Path with an Example Minimum Fixing Set

The fixing number of a path which consists of only a single vertex is 0 because such a path is a rigid graph. Any path on 2 or more vertices contains one reflection as its only symmetry. Thus, fixing one of the monovalent vertices of the path will fix the graph. Additionally, any single vertex of the path other than the center vertex (in a path on an odd number of vertices) will be a valid fixing set.

∎

2.5 Fixing Numbers of Complete Graphs

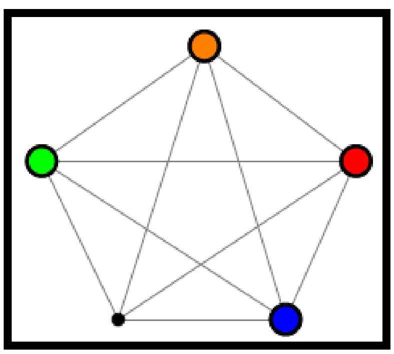


Figure 9 Complete Graph with an Example Minimum Fixing Set

The complete graph, K_n, is the unique graph on n vertices (up to isomorphism) in which every pair of vertices is connected by an edge [8].

Proposition 5.

$$\text{fix}(K_n) = n - 1 \qquad \textbf{Equation 7}$$

[2]

Proof:

Construct K_n with $n-2$ vertices fixed. Since K_n is vertex-transitive, this can be done without loss of generality. The automorphism group of K_n is S_n. As such, there exists an automorphism in $\mathrm{Aut}(K_n)$ which permutes the 2 unfixed vertices without affecting any of the other vertices. Thus, the set consisting of all but 2 of the vertices in K_n would not constitute a fixing set, as there would still be a remaining symmetry. Therefore,

$$\mathrm{fix}(K_n) = n - 1 \qquad \textbf{Equation 8}$$

■

2.6 Fixing Numbers of Complete Bipartite Graphs

A complete bipartite graph $K_{m,n}$ is a graph whose vertices are partitioned in to two non-empty sets such that every vertex in one set is adjacent to all vertices in the other set and no pair of vertices within the same set are adjacent [4].

Theorem 6.

$$fix\left(K_{1,1}\right) = 1 \qquad \textbf{Equation 9}$$

For all other (m, n),

$$\mathrm{fix}\left(K_{m,n}\right) = n + m - 2 \qquad \textbf{Equation 10}$$

[10].

Proof:

$K_{1,1}$ is the same graph as P_2, which was shown to have fixing number 1 by Theorem 4.

Let $n \geq 2$ and let

$$V(K_{m,n}) = A \cup B$$

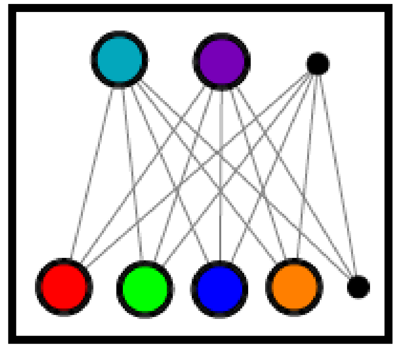


Figure 10 Complete Bipartite Graph with an Example Minimum Fixing Set

A and B are disjoint sets of vertices such that no 2 vertices within A are adjacent to eachother and no to vertices within B are adjacent to eachother. Fix m+n-3 vertices. At least 2 vertices in either A or B are not fixed and can be interchanged. This implies that

$$\text{fix}(K_{m,n}) > m + n - 3 \qquad \textbf{Equation 11}$$

Fix $(|A| = m) - 1$ vertices from A and $(|B| = n) - 1$ vertices from B. The 2 remaining vertices can not be interchanged. Therefore,

$$\text{fix}(K_{m,n}) = m + n - 2 \qquad \textbf{Equation 12}$$

■

An alternative proof that the fixing number of $K_{m,n}$ is $n + m - 2$ comes from examining $\overline{K_{m,n}}$.

Proof:

Let m and n not both equal 1. Consider $\overline{K_{m,n}}$, which is equal to $K_m \cup K_n$.

$$fix(K_m) = m - 1$$

$$fix(K_n) = n - 1.$$

Since m and n are not both equal to 1, K_m and K_n are not isomorphic rigid graphs. Employing **Equation 2** and **Equation 3**,we can see that

$$fix(\overline{K_{m,n}}) = (m - 1) + (n - 1) = m + n - 2 = fix(K_{m,n}) \qquad \textbf{Equation 13}$$

■

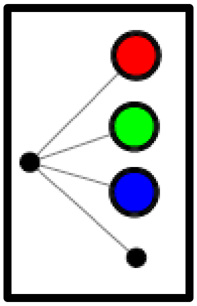


Figure 11 Star Graph

An interesting subclass of the complete bipartite graphs occurs when one of the partite sets has order 1 in the star graphs which are defined as $S_n = K_{1,n-1}$ [11]. In accordance with Equation 12,

$$\operatorname{fix}(S_n) = 1 + (n-1) - 2 = n - 2.$$

2.7 Fixing Numbers of Complete Multipartite Graphs

Complete multipartite graphs are a generalization of the underlying concept behind complete bipartite graphs A complete multipartite graph, $K_{m_1,m_2,\ldots,m_n}$, has its vertices partitioned into $n \geq 2$ non-empty sets such that no pair of vertices within a set are adjacent and all pairs of vertices in different sets are adjacent [12].

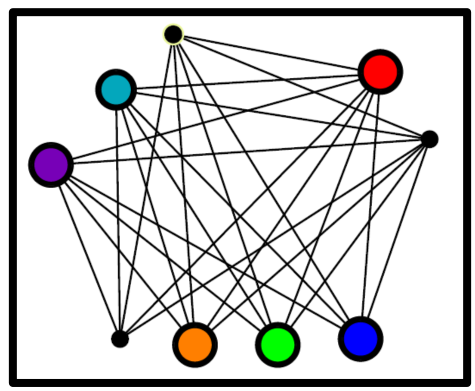


Figure 12 Complete Multipartite Graph with an Example Minimum Fixing Set

Lemma 7.

Consider the complete multipartite graph $K_{m_1,m_2,\ldots,m_n}$ with n non-empty partite sets, at most one of which consists of a single vertex of a single vertex. Then

$$fix(K_{m_1,\ldots,m_n}) = \sum_{i=1}^{n}(m_i - 1)$$

Equation 14

Proof:

Let at most one of $m_1, \ldots, m_n$ equal 1. Now consider $\overline{K_{m_1,\ldots,m_n}}$, which is congruent to $K_{m_1} \cup \ldots \cup K_{m_n}$. Since at most one of $m_1, \ldots, m_n$ equals 1, there are no isomorphic rigid connected components in $K_{m_1} \cup \ldots \cup K_{m_n}$. Thus,

$$\begin{aligned} \mathrm{fix}(K_{m_1} \cup \ldots K_{m_n}) &= \mathrm{fix}(K_{m_1}) + \cdots + \mathrm{fix}(K_{m_n}) = \\ &= (m_1 - 1) + \cdots + (m_n - 1) \end{aligned}$$

Equation 15

■

In the case where there are multiple partite sets, each of order 1, a slightly more sophisticated version of this formula is required to determine the fixing number. This is because the formula must account that even though a partite set of order 1 does not have pairs of vertices within it that can be interchanged with one another, entire partite sets of order 1 can be interchanged with one another, meaning that all but one of such partite sets must have their comprising vertex fixed.

Theorem 8.

Consider the complete multipartite graph $K_{m_1,m_2,\ldots,m_n}$ on n non-empty partitions, with $k > 1$ of the partitions consisting of a single vertex of a single vertex. Then

$$fix\left(K_{m_1,\ldots,m_n}\right) = k - 1 + \sum_{i=1}^{n}(m_i - 1)$$

Equation 16

Proof:

Consider $\overline{K_{m_1,\ldots,m_n}}$, which is congruent to $K_{m_1} \cup \ldots \cup K_{m_n}$. Since k of $m_1, \ldots, m_n$ equals 1, there are k isomorphic rigid connected components in $K_{m_1} \cup \ldots \cup K_{m_n}$, $k - 1$ of which must contain a single fixed vertex. Construct the complete multipartite graph H on $n - k$ partitions such that

$H + P_1 + \cdots + P_1 = G$ (with k copies of P_1 in the join) **Equation 17**

Thus,

$$\mathrm{fix}(G) = \mathrm{fix}(H) + (k - 1) = k - 1 + \sum_{i=1}^{n}(m_i - 1)$$

Equation 18

■

2.8 Fixing Numbers of Wheels

We define the wheel graph [13] as follows

$$W_n = C_n + P_1 \qquad \textbf{Equation 19}$$

This is the join of a cycle of order n with a single vertex.

Theorem 9.

$$\forall n \geq 4 \operatorname{fix}(W_n) = 2 \qquad \textbf{Equation 20}$$

Proof:

The symmetries of a wheel are rotation and reflection. In W_n, there are 2 orbits, one consisting of only the center vertex, and on consisting of all of the other vertices. Without loss of generality, select one of the vertices other than the center vertex to fix; call this vertex v. This removed rotational symmetry, but will not fix the graph, as reflection through the axis containing v will still be permissible. Therefore, the fixing number is greater than one. Now fix one of the vertices adjacent to v. This removes the remaining reflexive symmetry and thus fixes the graph. Therefore, the fixing number is 2. More generally, any 2 non-antipodal vertices of W_n will form a fixing set.

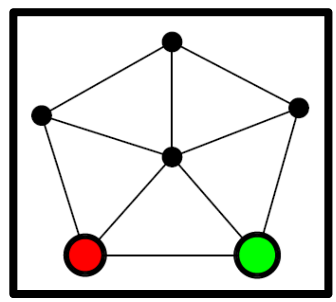


Figure 13 Wheel with an Example Minimum Fixing Set

■

We now generalize this result to the more general $K_1 + G$, for more general graphs than the cycles which were joined with K_1 to form the wheel graphs..

Theorem 10.

Let G be a graph of order n.

$$\begin{cases} \text{fix}(K_1 + G) = \text{fix}(G), & G \text{ has no vertices of degree } n-1 \\ \text{fix}(K_1 + G) = \text{fix}(G) + 1, & G \text{ has 1 vertex of degree } n-1 \end{cases}$$

Equation 21

Proof:

All of the automorphisms in G will necessarily be present in $G + K_1$. Supplementary to any of the automorphisms of G, if there is a vertex in G of degree $n-1$ then that vertex will be interchangeable with the vertex from K_1, as both vertices will have the same set of neighbors.

■

Theorem 11.

If G and H are 2 connected graphs with the order of G strictly less than that of H and additionally all of the vertices in H are of sufficiently small degree so that the following holds:

$$|V(G)| + \max_{v \in V(H)} \deg(v) < |V(H)| + 1$$

Equation 22

Then:

$$\text{fix}(G + H) = \text{fix}(G)\,\text{fix}(H)$$

Equation 23

Proof:

$G + H$ contains all of the automorphisms of both G and H; $\varphi_1 \in Aut(G)$ and $\varphi_2 \in Aut(H)$ can be applied sequentially to produce an automorphism in $Aut(G + H)$. Additionally, in the graph $G + H$ none of the vertices which had come from G will have the same degree as a vertex which had come from H, so there are no new automorphisms. Consequently, there will be no automorphisms of $G + H$ which include mapping a vertex that had come from G to a vertex that had come from H or vice versa. Thus we can apply the product rule to see that

$$\operatorname{fix}(G + H) = \operatorname{fix}(G)\operatorname{fix}(H)$$

■

2.9 Fixing Numbers of Friendship Graphs

Friendship graphs F_k were first defined by Erdös as graphs without C_4 in which every pair of vertices is connected by a path of length 2. This equates to a graph consisting of k triangles, which all share a common vertex and no 2 of which share a common edge [14].

Proposition 12.

$$\operatorname{fix}(F_k) = k \qquad \textbf{Equation 24}$$

Proof:

The center vertex is in its own orbit, and thus already has fixing number 1. The symmetries of F_n are the interchange of any non-center vertex v with the other non-center vertex that is on the same C_3 sub-graph as v. Additionally, any two C_3 sub-graphs can be interchanged. The only way to prevent the interchanging of two vertices on the same C_3 sub-graph is to fix one vertex on each C_3 sub-graph. This will additionally result in the removal of the symmetry in which two C_3 sub-graphs can be interchanged. Therefore, the fixing number is n.

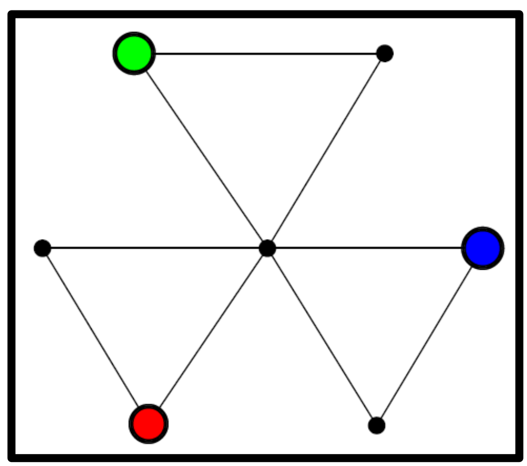


Figure 14 Friendship Graph with an Example Minimum Fixing Set

■

Friendship graphs can be viewed as a subset of the graphs defined by

$$G = K_1 + \bigcup_k H$$

Equation 25

where there are k copies of H and $H = P_2$.

We can state the preceding proposition more generally as follows:

Theorem 13.

Let H be any graph. We define the generalized friendship graph G with generator H by the following.

$$G = K_1 + \bigcup_k H$$

(a) If $\text{fix}(H) = 0$, then

$$\text{fix}\left(G = K_1 + \bigcup_k H\right) = k - 1$$

Equation 26

(b) If $\text{fix}(H) > 0$, then

$$\text{fix}\left(G = K_1 + \bigcup_k H\right) = k(\text{fix}(H))$$

Equation 27

Proof (a):

The k copies of H can all be interchanged. Thus at least one vertex from each copy of H except for one must be fixed. This will suffice to fix G as H by assumption has no automorphisms and the only automorphisms which must be removed from G are the interchanges of subgraphs isomorphic to H.

■

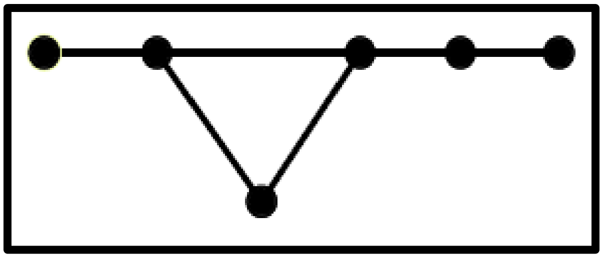


Figure 15 Generalized Friendship Graph Generator

Let H be the graph shown in Figure 15.

Consider the generalized friendship graph G with generator H and

$$G = K_1 + \bigcup_3 H$$

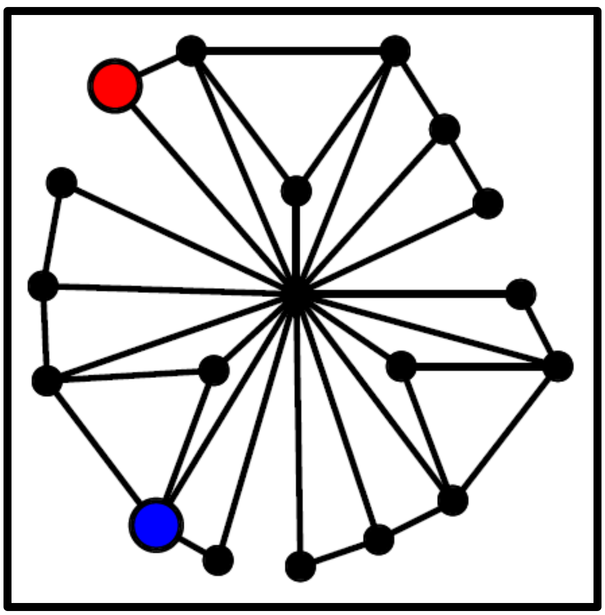


Figure 16 Generalized Friendship Graph with Rigid Generator

H is a rigid graph, but since the 3 subgraphs in G which are isomorphic to H can be mutually interchanged with one another, 2 of those subgraphs must contain a fixed vertex in order to fix G.

Proof (b):

The automorphisms of each instance of H in G must be removed. This will require fixing $\text{fix}(H)$ vertices from each of the k instances of H in G. This will suffice to fix G as each instance of H having one or more fixed vertices will prevent the copies from being interchanged with one another.

■

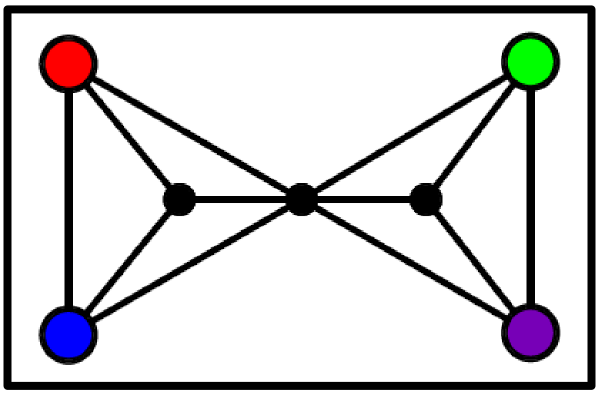


Figure 17 Generalized Friendship Graph with Non-rigid Generator

As an example, consider the graph

$$G = K_1 + \bigcup_{2} K_3$$

Here $H = K_3$, making H a non-rigid graph. There are 2 copies of H which are being joined to K_1, so $\text{fix}(G) = 2(2) = 4$

Chapter 2: Fixing Number Algorithms

The fixing numbers of many common classifications of special graphs were described in the preceding chapter. We will now develop a theory of the computation of fixing numbers for all graphs.

1 Tree Fixing Algorithms

A tree is defined as a connected graph with no cycles [8]. The leaf vertices of a tree are those vertices of degree 1. In [15], it was shown that every non-rigid tree of order greater than 1 has a minimum fixing set consisting of only leaf vertices. Consequently, computing the fixing number of a tree is a far easier problem than computing the fixing number of a graph in general. As such, we will examine these computations before considering the computations for more general graphs

1.1 Erwin and Harary's Algorithm

The first general computation of fixing number of trees was done by Harary and Erwin in [2]. Their original formulation phrased their computation as a theorem but it has been parsed out here into algorithmic form in order to make it easier to compare with the other algorithms that will be presented. Additionally, an effort has been made to simplify the terminology used in the original publication.

Consider a tree T

Let $V' \subset V(T)$ denote the set of leaf vertices in T

Construct the digraph T' with $V(T') = V(T)$

For each leaf vertex $v \in V'$, create the directed edge (v, x) in T'

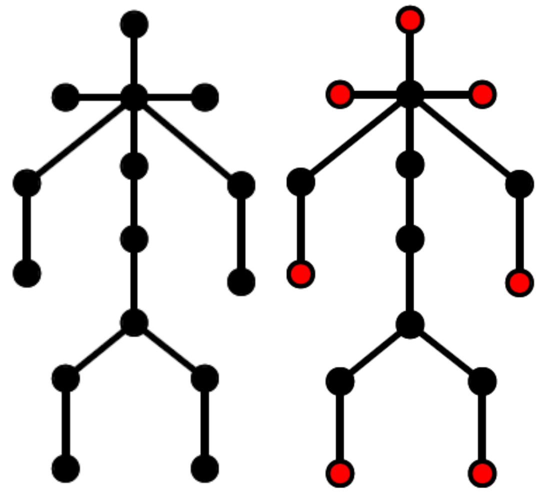
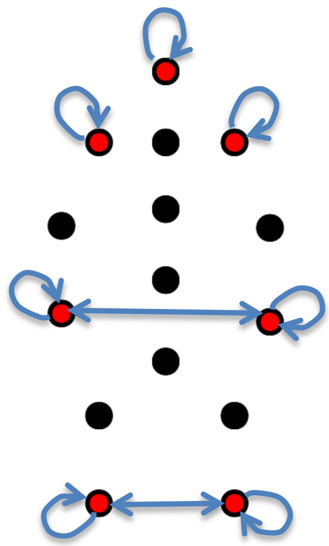
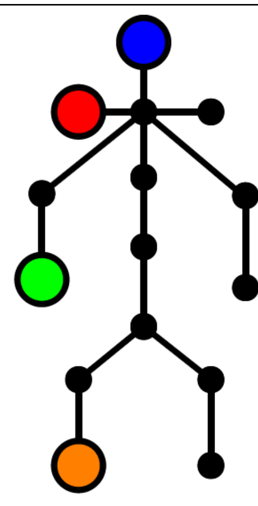
iff $\forall \varphi \in Aut(T), x = \varphi(x) \rightarrow v = \varphi(v)$

The fixing number of T is the order of the smallest subset S of $V(T')$ such that there is at most one leaf from each orbit of T that does not have an edge directed towards it from a vertex in S

Algorithm 1 Erwin and Harary Tree Fixing Algorithm

This algorithm essentially combines the result that every non-rigid tree has a minimum fixing set consisting of exclusively leaf vertices [2] and that in order for a graph to be fixed, every vertex in that graph must be fixed in order to provide an algorithm which is marginally better than the naïve approach of testing all possible sets of vertices to determine if the fix the graph. We will now proceed through a sample execution of this algorithm in order to aid in the reader's understanding of the algorithm.

1.1.1 Example

Identify the leaf vertices of the tree T.	
Construct the digraph T'.	
The fixing number of T is the order of the smallest subset S of $V(T')$ such that there is at most one leaf from each orbit of T that does not have an edge directed towards it from a vertex in S. In this example, the fixing number is 4.	

1.2 Minimum-Determining-Set-Tree Algorithm

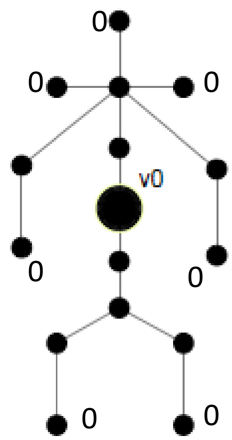
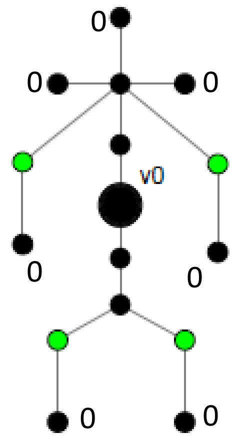
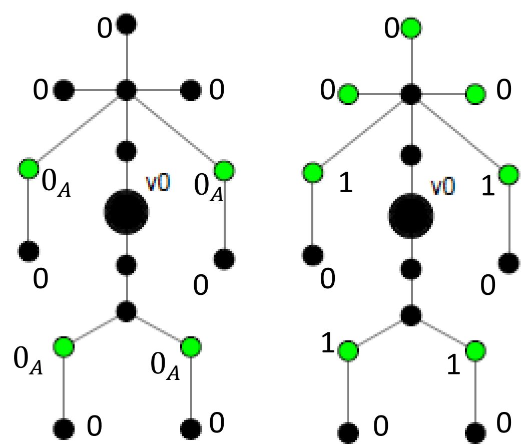
The minimum-determining-set-tree algorithm was presented by Cáceres et. al. in 2010 [15]. It is based on the result that every tree contains a minimal fixing set consisting entirely of leaf vertices [2], [15].

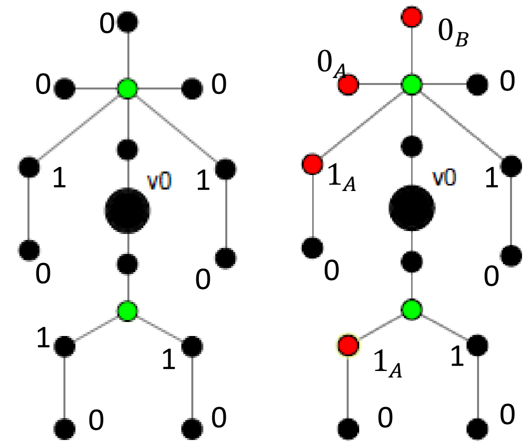
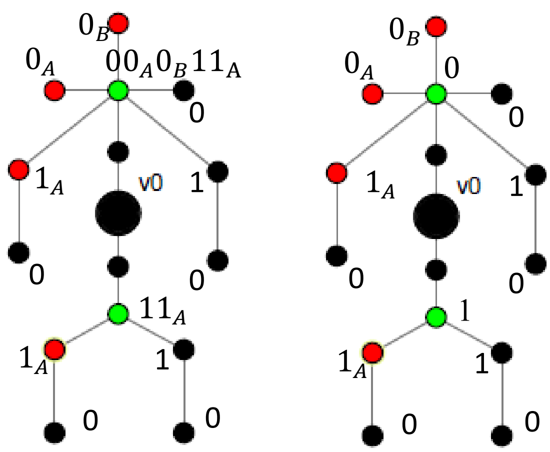
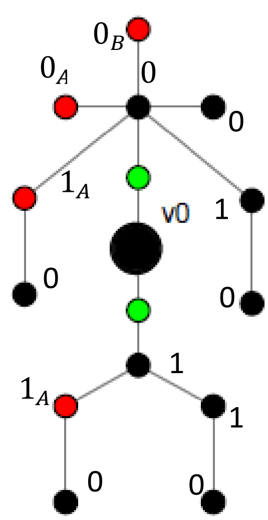
Compute the center of the tree.
Consider the version of this tree which is rooted at its center.
If the center consists of 2 vertices, insert a new vertex in between them to serve as the center.
Assign label "0" to all of the leaf vertices.
For $i \coloneqq$ (radius – 1) to 1, with i decreasing by 1 each iteration
  For each non-leaf vertex v at distance i from the root
    Look at the sets of children of that vertex with the same label. The children of a vertex v are those vertices which are adjacent to v and have greater distance from the center than v. Fix all but one from each set, giving each fixed vertex its own label.
    Label v with the lexicographically ordered concatenation of the labels of its children.
    Lexicographically order all of the vertices at distance i from the radius and re-label them with the
      non-negative numbers, mapping one natural number to each label.

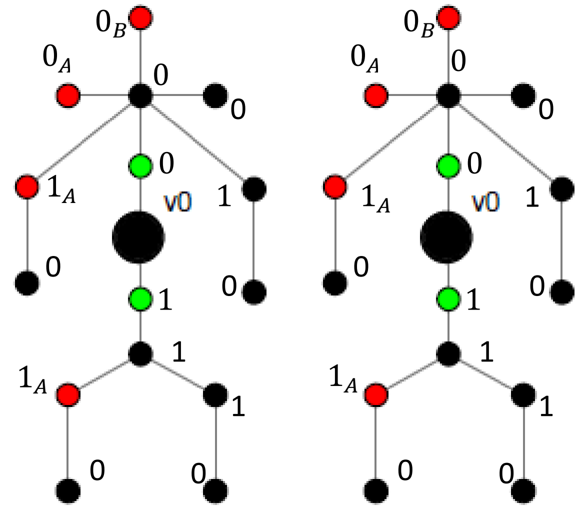
Algorithm 2 Minimum Determining-Set Tree Algorithm

1.2.1 Example

In this example, green will be used to highlight the vertices that are currently under consideration and red will be used to denote vertices that have been fixed.

The center of this tree was an edge, so we have inserted the new vertex v_0 to act as the center. Note that the radius of this tree is 4. Assign label 0 to all leaf vertices.	
Consider all of the vertices at distance 3 from the center. For each vertex under consideration, fix all but one of its children with the same label. In this example at this step, no vertices will be fixed because each vertex has only one child.	
Label each vertex that is at distance 3 from the center with the lexicographically ordered concatenation of the labels of its children. Relabel the vertices at distance 3 from the center with the non-negative numbers.	

Consider all of the vertices at distance 2 from the center. For each vertex under consideration, fix all but one of its children that have the same label.	
Label each vertex that is at distance 2 from the center with the lexicographically ordered concatenation of the labels of its children. Relabel the vertices at distance 2 from the center with the non-negative numbers.	
Consider all of the vertices at distance 1 from the center. For each vertex under consideration, fix all but one of its children with the same label. In this example at this step, no vertices will be fixed.	

Label each vertex that is at distance 1 from the center with the lexicographically ordered concatenation of the labels of its children. Relabel the vertices at distance 1 from the center with the non-negative numbers. The graph is now fixed. The fixing number is 4 and the red vertices form an example of a minimum fixing set.	

1.3 Contraction Tree Fixing Algorithm

We first define a weight preserving subdivision as an action on a graph which takes as input a weighted graph one or more pairs of vertices from that graph along with the weights of the edges connecting those vertices. The edge between each of those pairs of vertices is replaced with a weighted path such that the sum of the weights of the edges along the newly added graph is equal to the weight of the edge which was removed.

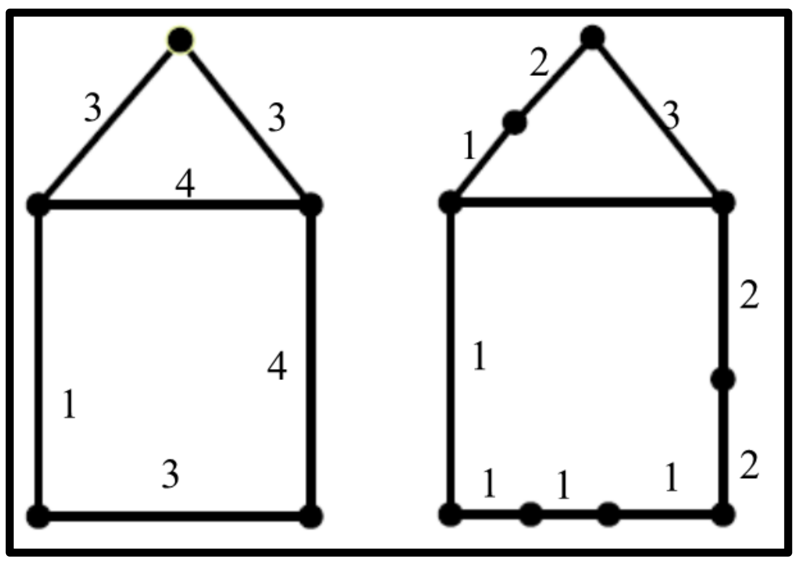


Figure 18 Weight Preserving Subdivision

The following algorithm is an equivalent alternative to the previous algorithm. It is an original alternative that is slightly more efficient for trees which have many vertices of degree 2 because of a relatively simple pre-processing step.

Let T be a tree, not a path. (If T is a path, it's fixing number is 1.)

Give each edge in T weight 1.

Consider T', with no vertices of degree 2, of which T is a weight-preserving subdivision.

If T has no vertices of degree 2, then $T = T'$.

Let S be the set of vertices in T' which are adjacent to a vertex of degree 1.

For each $v \in S$, let $D(v)$ be the set of weights of the pendant edges incident with v.

Let $\deg_k(v)$ be the number of edges of weight k which are incident to v.

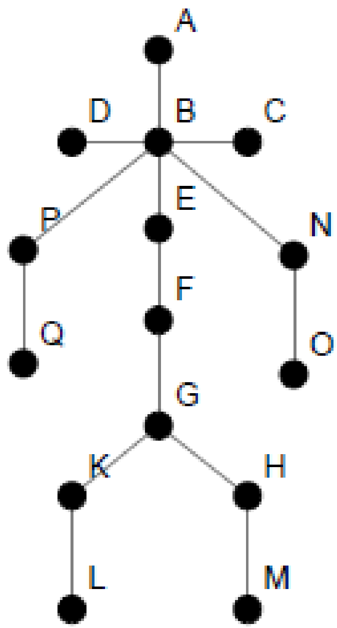
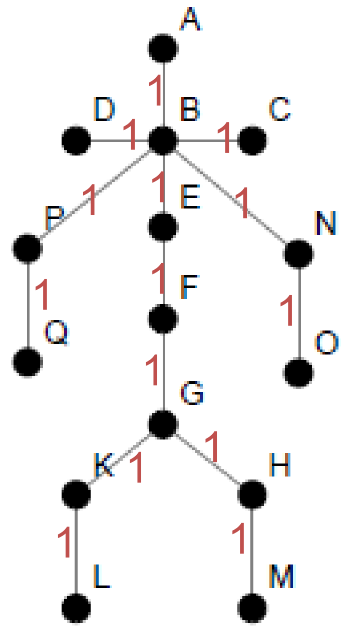
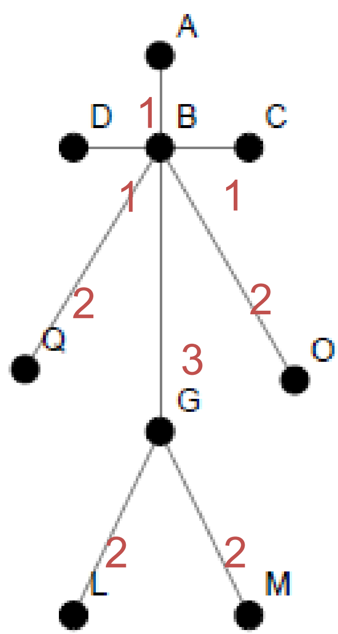
Calculate $\deg_k(v)$ for all $v \in S$, for all $k \in D(v)$

$$Fix(T) = \sum_{v \in S} \sum_{k \in D(v)} (deg_k(v) - 1)$$

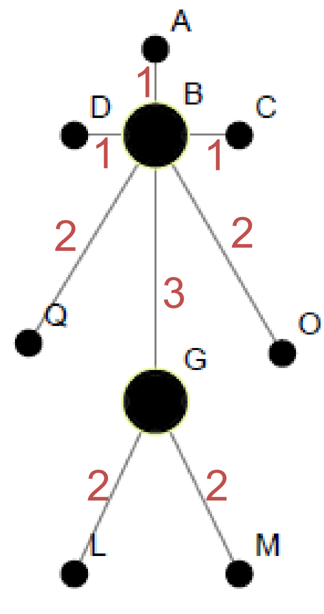
Algorithm 3 Contraction Tree Fixing Algorithm

The following is an example of this algorithm as applied to a specific graph so as to aid the reader in following the execution process.

1.3.1 Example

Let T be a tree, not a path. Give each edge in T weight 1.	 
Construct T'	

<table>
<tr>
<td>Let S be the set of vertices in T' which are adjacent to a vertex of degree 1.

Calculate $D(v)$ for each $v \in S$.</td>
<td>

$S = \{B, G\}$

$D(B) = \{1, 2\}$
$D(G) = \{2\}$</td>
</tr>
<tr>
<td>Calculate $deg_k(v)$ for all $v \in S$, for all $k \in D(v)$.</td>
<td>$\deg_1(B) = 3$
$\deg_2(B) = 2$

$\deg_2(G) = 2$</td>
</tr>
<tr>
<td>$$Fix(T) = \sum_{v \in S} \sum_{k \in D(v)} (deg_k(v) - 1)$$</td>
<td>$Fix(T) = (3 - 1) + (2 - 1) + (2 - 1) = 4$</td>
</tr>
</table>

1.3.2 Proof of Correctness

The symmetry group of a tree is the direct product of wreath products of isomorphic branches connected at a common vertex. We define a branch as a proper subgraph for which there are no 2 vertices in the same component of the branch but not members of the branch themselves, which have a path connecting them that includes one or more vertices of the branch. Additionally, there will be exactly one vertex of the branch which is adjacent to vertices not in the branch and this vertex will have degree at least 3. In order to remove all but the trivial automorphism, at least 1 vertex from each branch which is isomorphic to another branch must be fixed. One branch in each set of isomorphic branches does not need to have a vertex fixed because it will be the only such branch in that set. This is exactly the set of vertices which are fixed by the above algorithm

■

1.3.3 Efficient Construction of the Homeomorphically Irreducible Tree

In order for the Contraction Tree Fixing Algorithm to be a viable alternative the Minimum Determining Set Tree Algorithm, it is essential that computation not be wasted during the conversion from a possibly homeomorphically reducible tree to a homeomorphically irreducible tree. We employ the use of carefully selected data structures for storing the tree to do just that, namely by storing the tree with an index on the degree of vertices. Our homeomorphic reduction algorithm proceeds as follows.

```
For each vertex of degree 2
        Consider the vertices, v and u, which are adjacent to it
        Create the edge u,v (with arc)
        Delete both of the edges incident with the current vertex under consideration
```

Algorithm 4 Homeomorphic Reduction Algorithm

This algorithm will clearly require a small constant time for each vertex of degree 2, and with indexing of the tree data structure, can execute overall in time linear with the number of vertices of degree 2.

1.4 Comparison of Tree Fixing Algorithms

The Erwin and Harary algorithm was aptly described as a theorem and not an algorithm. While a corresponding algorithm does immediately follow from reading their work, that algorithm is much more a matter of exhaustive search relative to the definition of fixing number rather than anything which resembles true algorithmic flavor and the corresponding efficiency that one would like to see from such an algorithm. That said, there were some features of their work, such as the intermediate construction of a digraph that do provide substantial gains over a truly naïve algorithm which would just check all sets of vertices in order to find the smallest set which fixes the graph. Overall though, this work provided a great foundation in its own right, but in comparison to the other available algorithms doesn't merit being used for any trees of non-trivial size.

Though discovered independently, the Contraction Tree Fixing Algorithm is essentially a modification of the Minimum Determining Set Tree Algorithm. Both algorithms can run in linear time [15] in the number of vertices and for the most part execute in very similar manners. The Contraction Tree Fixing Algorithm has a preprocessing step that will take no time in homeomorphically irreducible trees, will take minimal time in homeomorphically reducible trees and will result in faster execution of the main algorithm in homeomorphically reducible trees. Once this pre-processing step is complete, the algorithms are identical. For homeomorphically reducible trees, the Contraction Tree Fixing Algorithm will run slightly faster because the time required for each vertex of degree 2 to delete the edges incident with it and insert a new edge is shorter than the time required for labeling each of the vertices by the Minimum Determining-Set Fixing Algorithm. The gains made by pre-processing in the Contraction Tree Fixing Algorithm, while existent are minimal and do not warrant not using the Minimum Determining Set Tree Algorithm if there are other reasons to do so.

2 General Graph Fixing Algorithms

We will now turn our attention to algorithms which are designed to compute the fixing number of any graph. These algorithms can be applied to trees, but will be far more inefficient than the algorithms designed specifically for fixing trees.

2.1 Greedy Fixing Algorithm

Gibbons and Laison [1]proposed the following algorithm as an attempt at an algorithm capable of finding the fixing number of any graph. This has come to be known as *the* Greedy Fixing Algorithm. This is a constructive algorithm, which in addition to returning the fixing number of the graph, determines a corresponding fixing set as a side-effect. As the name implies, the algorithm employs a greedy approach, iteratively fixing a vertex from the largest remaining orbit until the entire graph is fixed. Consequently, the algorithm is clearly guaranteed to construct a minimal fixing set, but as we will see below it is not necessarily minimum.

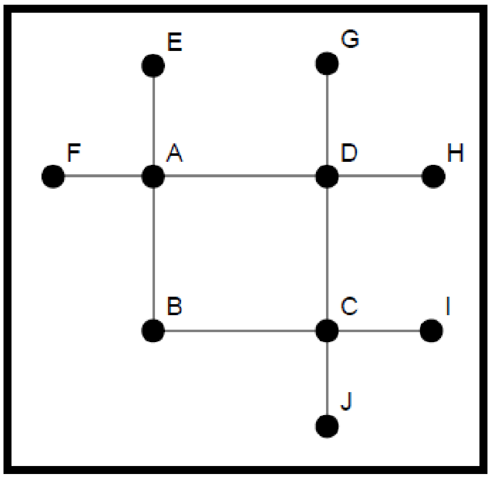
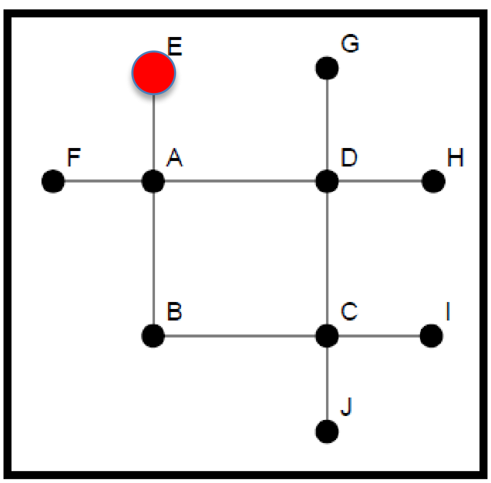
The Greedy Fixing Algorithm proceeds as follows:

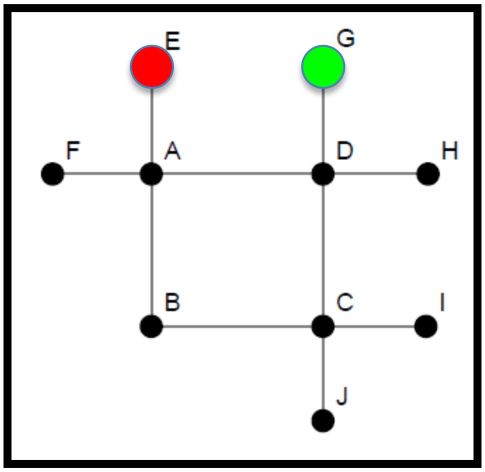
Find a largest orbit in G
Fix a vertex in a largest orbit of G
Repeat the first step until all orbits are of size 1

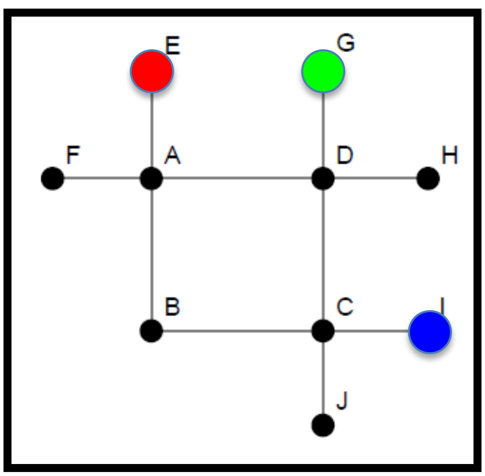
Algorithm 5 Greedy Fixing Algorithm

2.1.1 Example

For illustrative purposes, we will now go through an example execution of the Greedy Fixing Algorithm.

Graph	Orbits
	$\{E, F, I, J\}$ $\{A, C\}$ $\{G, H\}$ $\{B\}$ $\{D\}$ The largest orbit is $\{E, F, I, J\}$, so one of the vertices from that orbit will be fixed in the next iteration.
	$\{G, H\}$ $\{I, J\}$ $\{A\}$ $\{B\}$ $\{C\}$ $\{D\}$ $\{E\}$ $\{F\}$ Both of the orbits $\{G, H\}$ and $\{I, J\}$ are of largest size. In the next iteration, we will select at random one of the vertices from one of these orbits to fix.

<table>
<tr>
<td>
</td>
<td>$\{I, J\}$
$\{A\}$
$\{B\}$
$\{C\}$
$\{D\}$
$\{E\}$
$\{F\}$
$\{G\}$
$\{H\}$

The largest orbit is $\{I, J\}$, so we will select one of these vertices to fix in the next iteration.</td>
</tr>
<tr>
<td>
</td>
<td>$\{A\}$
$\{B\}$
$\{C\}$
$\{D\}$
$\{E\}$
$\{F\}$
$\{G\}$
$\{H\}$
$\{I\}$
$\{J\}$

At this point, all of the orbits are of size 1, so the graph is fixed. The fixing number is 3 and one possible minimum fixing set is $\{E, G, I\}$.</td>
</tr>
</table>

2.1.2 Computational Complexity

The computational bottleneck in the Greedy Fixing Algorithm is that it requires computing all of the orbits of the graph and repeating this process for each fixed vertex. Fixing a vertex changes the orbits of the graph, but not in a way which can be easily calculated from knowing which vertex was fixed and what the orbits of the graph were prior to fixing that vertex. Thus the computations required to determine a largest orbit of the graph must be done after each vertex is fixed without any reuse of computation. There are some computational shortcuts that can be taken in calculating the orbits and the process can be greatly sped up by choosing appropriate mathematical software and admittedly not all orbits need be found if one can prove that they have found the largest, but this still remains a huge computational challenge. Additionally, most of the known methods for computing orbits are either sufficiently close to a naïve method so as to fail to yield substantial computational gains or serve only to provide approximations of the orbits, something which we cannot allow for in our algorithms. The interested reader can find some of the methods for computing orbits in [16].

2.1.3 Counter-Example

We now present the following counterexample to the greedy fixing algorithm proposed by [1]. It is a counter example in that the initial selection of vertices to fix (within the bounds allowed by the algorithm) can result in different answers being produced by the algorithm. The actual fixing number of this graph is 3, but the greedy algorithm can produce both 3 and 4 as results. We propose that this is a minimal counter example in terms of number of vertices.

As can be seen in Figure 19, this graph contains 2 orbits, each of order 6. We shall henceforth refer to these orbits as the red and blue orbits for convenience.

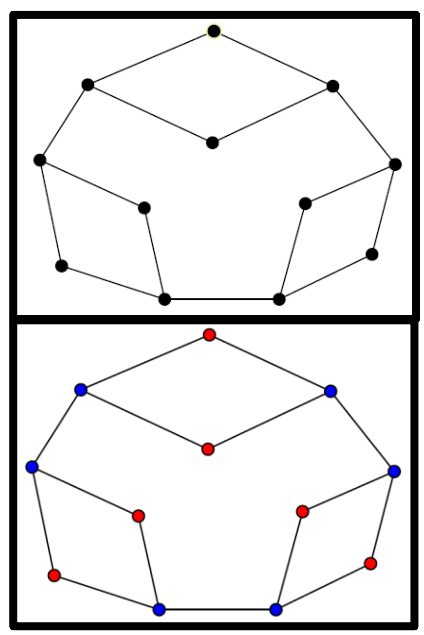


Figure 19 Counter-Example to the Greedy Fixing Algorithm

Since both orbits are of the same size, the initial vertex chosen to be fixed can be from either orbit.

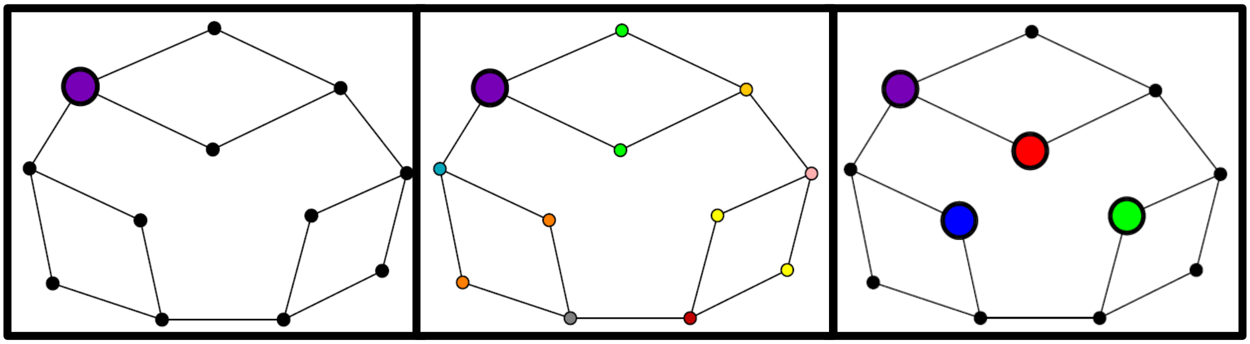


Figure 20 One Possible Execution of the Greedy Fixing Algorithm

If one of the vertices from the blue orbit is initially fixed, the initial result will be for the graph to contain 3 orbits of order 2 and 6 orbits of order 1. The remaining 3 iterations of the algorithm will each include fixing one of the vertices from an orbit of order 2. Each time this is done, the effect on the orbits of the graph will be to decrease the number of orbits of order 2 by 1 and to increase the number of orbits of order 1 by 2. Thus this execution of the algorithm will produce a minimal fixing set of order 4. However, as we will see, this set is not minimum.

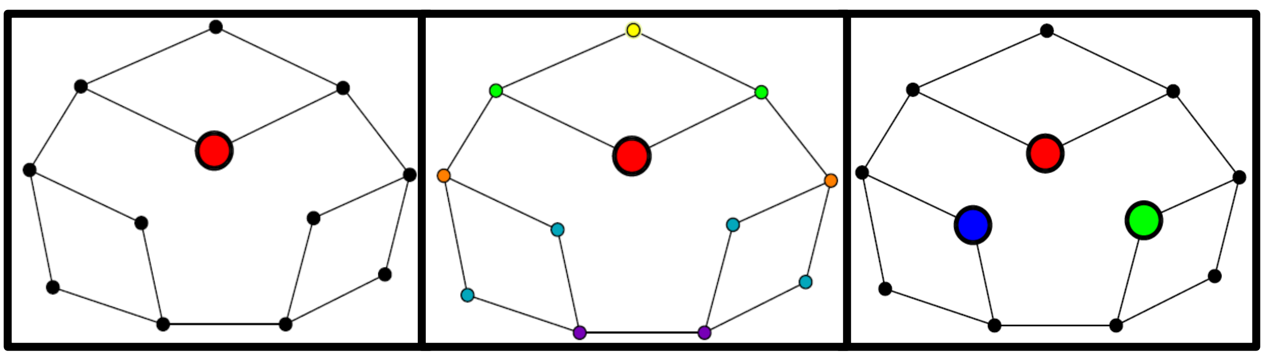


Figure 21 A Second Possible Execution of the Greedy Fixing Algorithm

Alternatively, if one of the vertices from the red orbit is fixed first, the initial result will be for the graph to contain 1 orbit of order 4, 3 orbits of order 2, and 2 orbits of order 1. The next iteration of the algorithm will fix one of the vertices from the orbit of order 4, which will leave the graph with 1 orbit of order 2 and 10 orbits of order 1. One of the vertices from the orbit of order 2 will then be fixed, leaving the graph with exclusively orbits of order 1 and producing a minimal fixing set of order 3. Note that this is actually a minimum fixing set.

2.2 Derangement Fixing Algorithm

We have developed an alternative to the greedy fixing algorithm as follows.

Erwin and Harary defined a (u, v) −interchange, φ, as an automorphism in $Aut(G)$ with $u, v \in V(G)$ such that φ transposes u and v and does not affect any of the other vertices of the graph [2].

We extend the concept of (u, v) −interchanges to apply to sets of vertices of any size. A derangement is a permutation in which no value is mapped to itself [17]. Given a graph G, we define a k-subset S of $V(G)$ to be an **S-derangement** of order $|S|$ if there exists an automorphism of G that deranges all of the vertices in S and fixes all of the vertices not in S, and there is no proper subset of S for which this condition holds. Note that unlike Erwin and Harary, we are using the term to refer to the set of vertices which are deranged, not to the automorphism which deranges them. This is necessary because in S-derangements where $|S| > 2$, there are potentially many automorphisms which derange all of the vertices in S while fixing all other vertices. An example S-derangement can be seen in Figure 22.

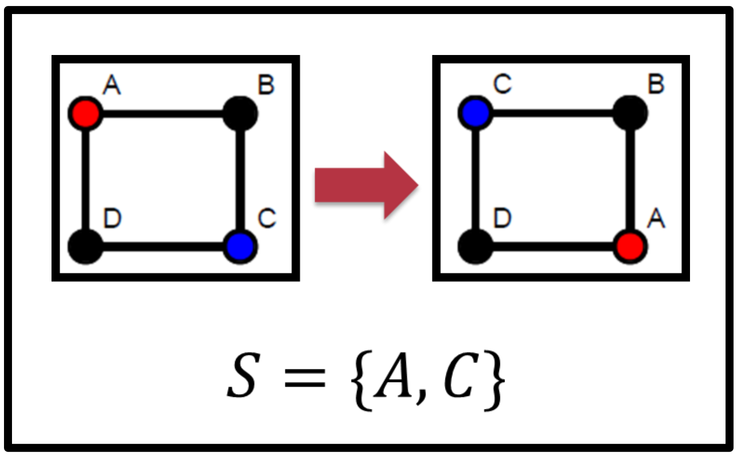


Figure 22 Example S-Derangement

Algorithm:

Let G be a graph.

Consider the (possibly non-disjoint) sets of vertices, $S_0, \ldots, S_k$, that comprise all S −derangements of G.

Compute the minimum set $X \in V(G)\ s.t.\ \forall 0 \leq i \leq k, X \cap S_i \neq \emptyset$.

$\mathrm{fix}(G) = |X|$

X is a minimum fixing set of G.

Algorithm 6 Derangement Fixing Algorithm

2.2.1 Proof of Correctness

By definition of a fixing set, at least one vertex in each fixing set must be fixed. If fixing one vertex from each fixing set was not sufficient to fix the graph, then there would consequently exist a set of vertices which could be permuted by an automorphism on G while fixing the remaining vertices in the graph, which includes at least one vertex from each fixing set. This set of vertices would comprise a fixing set, which is a contradiction. Therefore, the S-derangement algorithm correctly computes the fixing number of a graph.

■

2.2.2 Examples

The following two figures show examples of execution of the S-Derangement Fixing Algorithm.

In the first example there are 3 overlapping derangements, consisting of 3 vertices in total. Selecting 2 of these vertices to include in the fixing set suffices to have at least one vertex selected from each of the derangements. Additionally there is a derangement of order 4 that doesn't have any overlapping vertices with the other derangements. Consequently, one vertex must be chosen to be in the fixing set from the set of vertices comprising this derangement. The produces a fixing number of 3 for this graph.

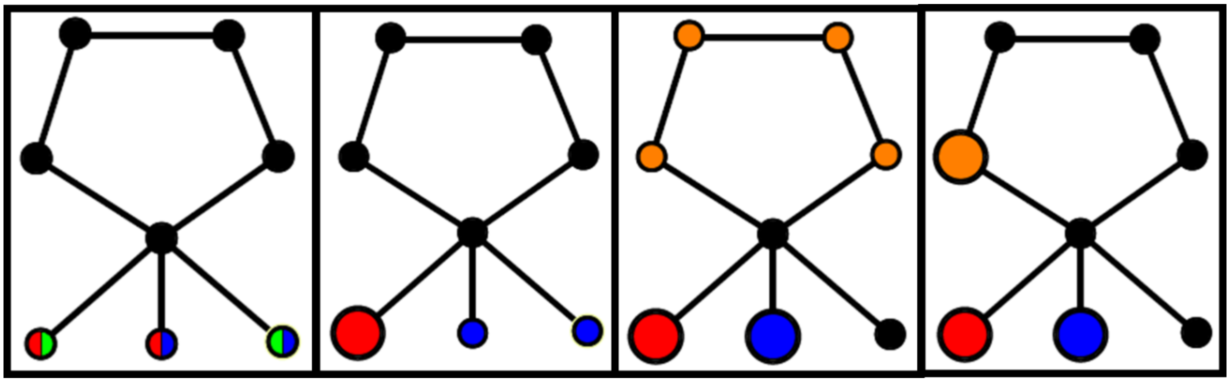


Figure 23 Example 1 of the Derangement Fixing Algorithm

The next example is included not because it is particularly illuminative of the process of the S-Derangement Fixing Algorithm, but rather to illustrate that the counter-example that we found to the Greedy Fixing algorithm can be accurately handled by this algorithm

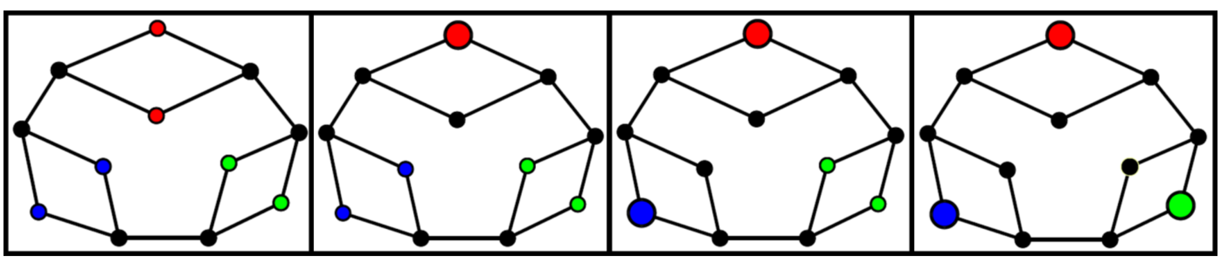


Figure 24 Example 2 of the Derangement Fixing Algorithm

2.2.3 Computational Considerations

The determination of S-derangements on trivially sized graphs is a relatively easy task for a human to perform, but is deceptively difficult, as it implicitly requires examining orbits, which we have already seen to be a daunting task.

In identifying the S-derangements of a graph, one must begin by examining the automorphism group of the graph, an admittedly complex procedure. There is an onto relationship from the prime subgroups of the automorphism group to the S-derangements. The exact vertices which constitute each S-derangement can be determined

Chapter 3: Distribution of Fixing Numbers

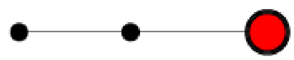
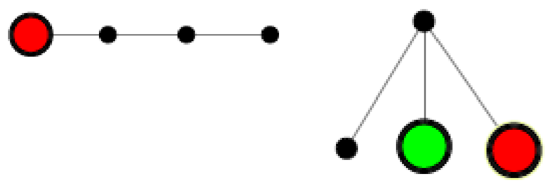
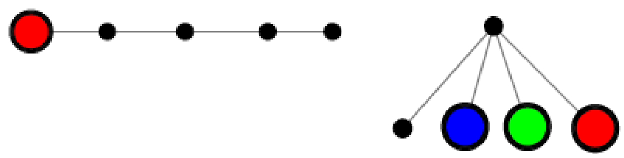
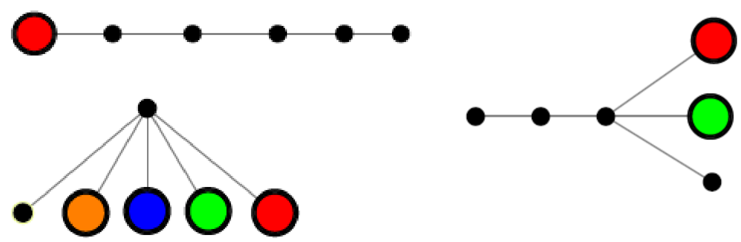
Historically, the distinction between the level of symmetry in different graphs has been denoted either by the automorphism group or by the rigidity of the graph. The fixing number of a graph provides a metric for quantifying the amount of symmetry in a graph and refining the distinction between rigid and non-rigid graphs. In order to use this metric to compare the general symmetries of different classes of graphs, we now turn our attention to studying the distribution of fixing numbers of graphs of a given type and given order.

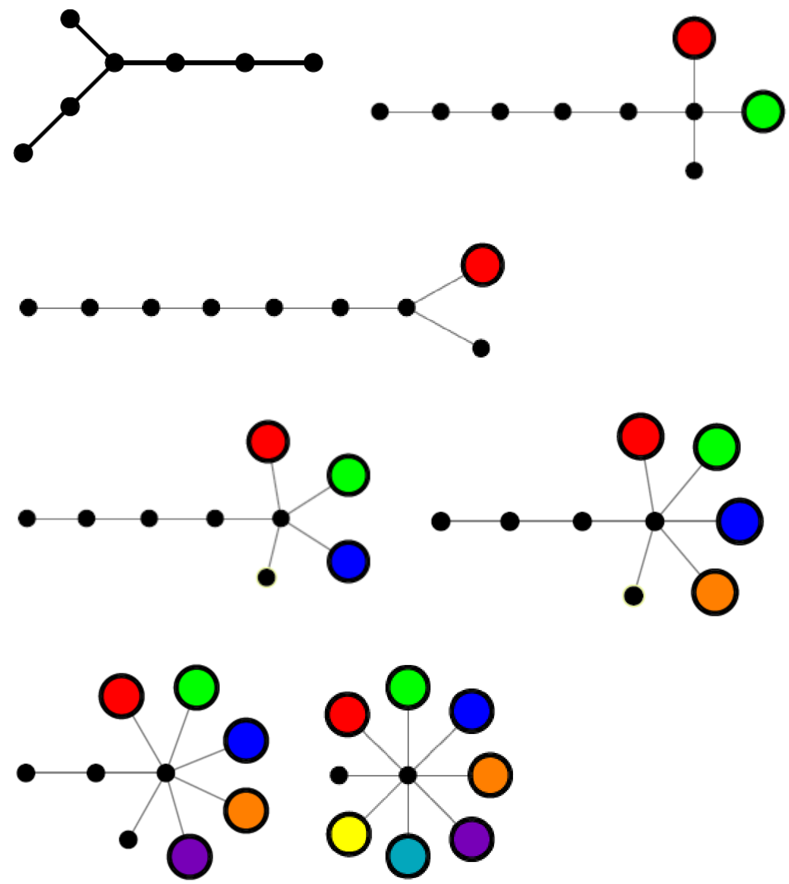
1 Distribution of Fixing Numbers of Trees

Trees form an interesting class of graph to study the distribution of fixing numbers because of the fact that most trees are non-rigid [7].

1.1 Possible Fixing Numbers

The following chart shows, by example, the possible fixing numbers for trees of small order. Note that the graphs shown in this chart serve to provide examples of trees with the possible fixing numbers, but this is not an exhaustive listing of small trees. Such lists can be found in [18]. The possible fixing numbers for larger trees are then given by an explicit proof.

n	Possible Fixing Numbers	Sample Graphs
1	0	
2	1	
3	1	
4	1, 2	
5	1, 3	
6	1, 2, 4	

n	Possible Fixing Numbers	Sample Graphs
≥7	0, 1, 2, …, n-5, n-4, n-2	

Theorem 14.

The fixing number of a tree with $n \geq 7$ vertices can be any value from 0 to $n-2$ other than $n-3$.

This theorem and a corresponding proof were published in [15], but we had independently discovered the proof provided below.

Proof:

It is well established that there exist rigid (i.e. with fixing number 0) trees of order n for all $n \geq 7$. Trees of order n with fixing number 1 to $n-2$, except $n-3$ can easily be constructed as stars with 1 branch being longer than a path of length 2. For example, see the section for $n \geq 7$ in the chart above. Assume that there is a tree with order $n \geq 7$ and fixing number $n - 3$. Such a tree would have to conform to one of the following cases:

1. 1 orbit of $n - 2$ vertices and 2 orbits of 1 vertex each.
2. 2 distinct orbits with $n - 1$ vertices in total and more than 1 vertex each and 1 orbit of 1 vertex.
3. 3 distinct orbits with n vertices in total and more than 1 vertex each.

In case 1, for each of the 2 vertices not in the largest orbit, the $n - 2$ vertices in the largest orbit would have to either all be adjacent or all be not adjacent to that vertex. The 2 vertices which each comprise their own orbit are either adjacent to each other or not; in either case, these vertices will be interchangeable (and thus in the same orbit) if they are both adjacent to the vertices in the larger orbit or both not adjacent to the vertices in the largest orbit. Thus, one of them must be adjacent to all of the vertices in the larges orbit, and the other not adjacent. If the 2 vertices which comprise their own orbits are not adjacent, then the graph will not be connected and thus not a tree. However if the vertices are connected, then the single vertex which was defined to comprise its own orbit and not be adjacent to any of the vertices in the largest orbit, would itself become a member of that orbit, which is a contradiction. Therefore, case 1 cannot exist. In case 2, for each of the larger orbits, the single vertex which comprises its own orbit would have to either be adjacent to all of the vertices in the orbit or not adjacent to any of the vertices in the orbit. Additionally, the vertices within an orbit cannot be adjacent to each other, as this would create cycles. If the 2 larger orbits both have all vertices adjacent to the single vertex which is its own orbit, then both of the larger orbits will actually comprise 1 orbit, which is a contradiction. Alternatively, if 1 of the larger orbits has all vertices not adjacent to the single vertex, then the graph would not be connected, and consequently not a tree. Therefore, case 2 cannot exist.

In case 3, in order for the graph to be connected the vertices in one of the orbits must each be adjacent to all of the vertices in another orbit. This would create cycles, which would be a contradiction. Therefore, case 3 cannot exist.

Therefore, there is no tree with order $n \geq 7$ and fixing number $n - 3$.

■

Corollary15.

If a graph has order ≥ 7 and it's complement is a tree, then the fixing number of that graph is not $n - 3$.

Proof:

This follows directly from combining Theorem 1 and Theorem 14.

■

1.2 Distribution of Fixing Numbers of Trees with up to 12 Vertices

We have calculated the fixing numbers for all trees up to order 12. The results of this enumeration for trees of order 9, 10, 11, and 12 are shown in the set of charts below. For trees of order 9, the most common fixing number is 1, while for trees of order 10, 11, or 12 the most common fixing number is 2. As n tends towards ∞, the most common fixing number of trees of order n will also tend towards ∞.

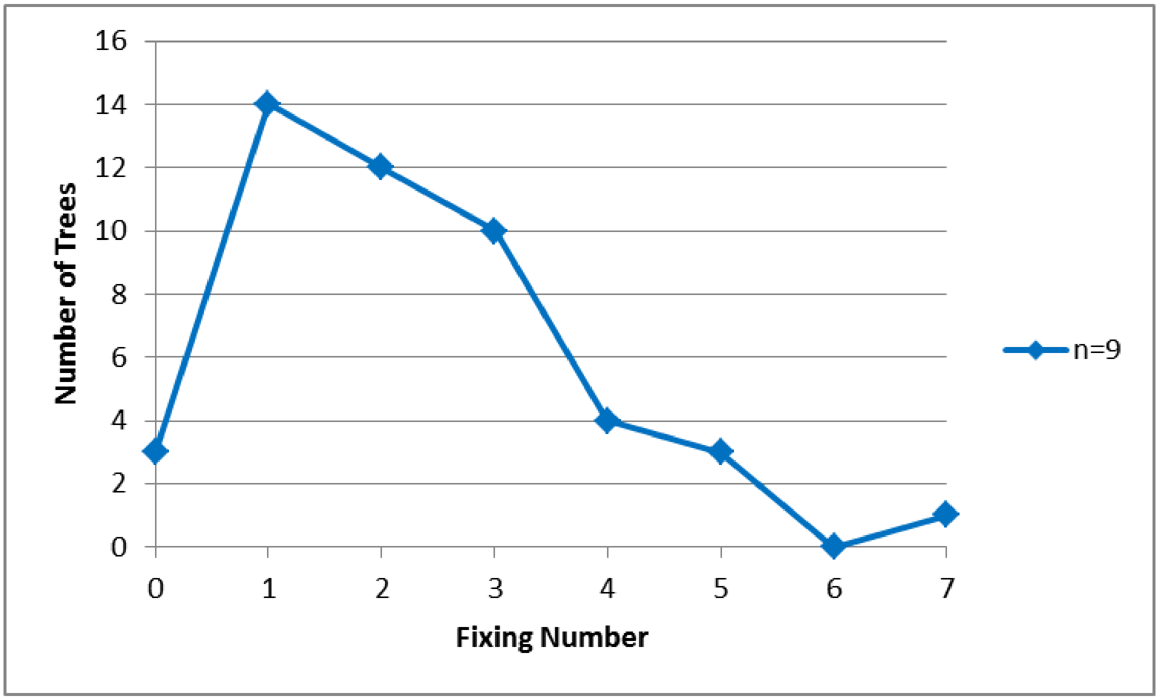


Figure 26 Distribution of Fixing Numbers of Trees of Order 9

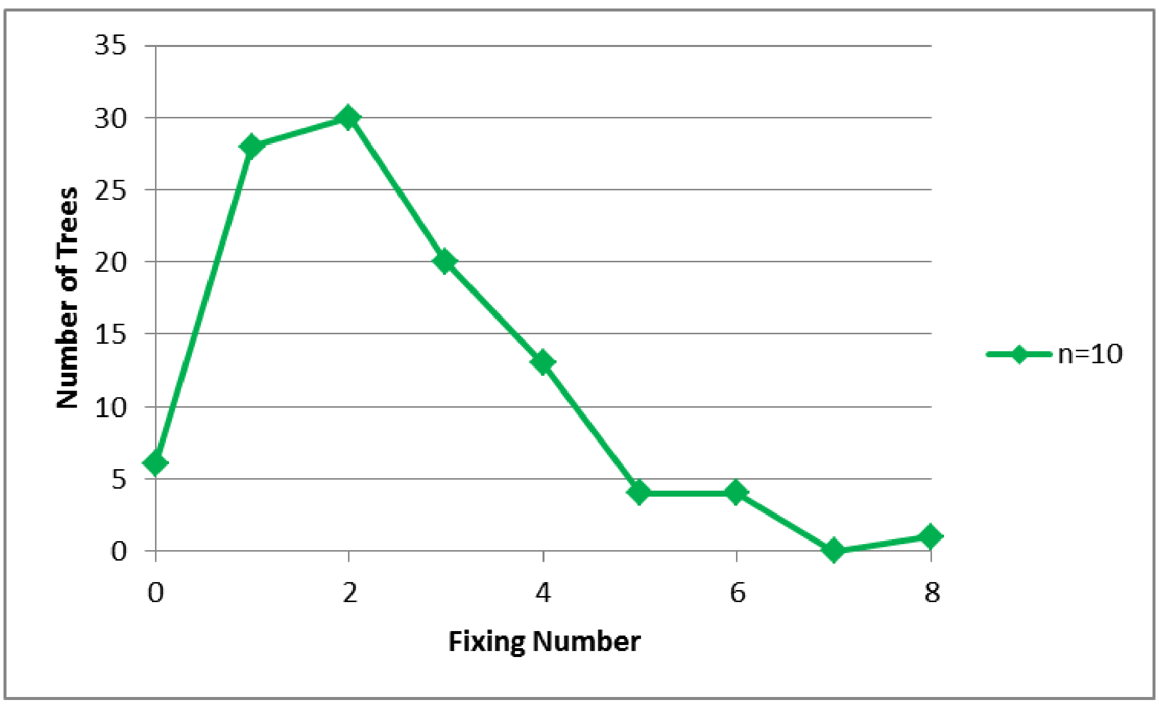


Figure 25 Distribution of Fixing Numbers of Trees of Order 10

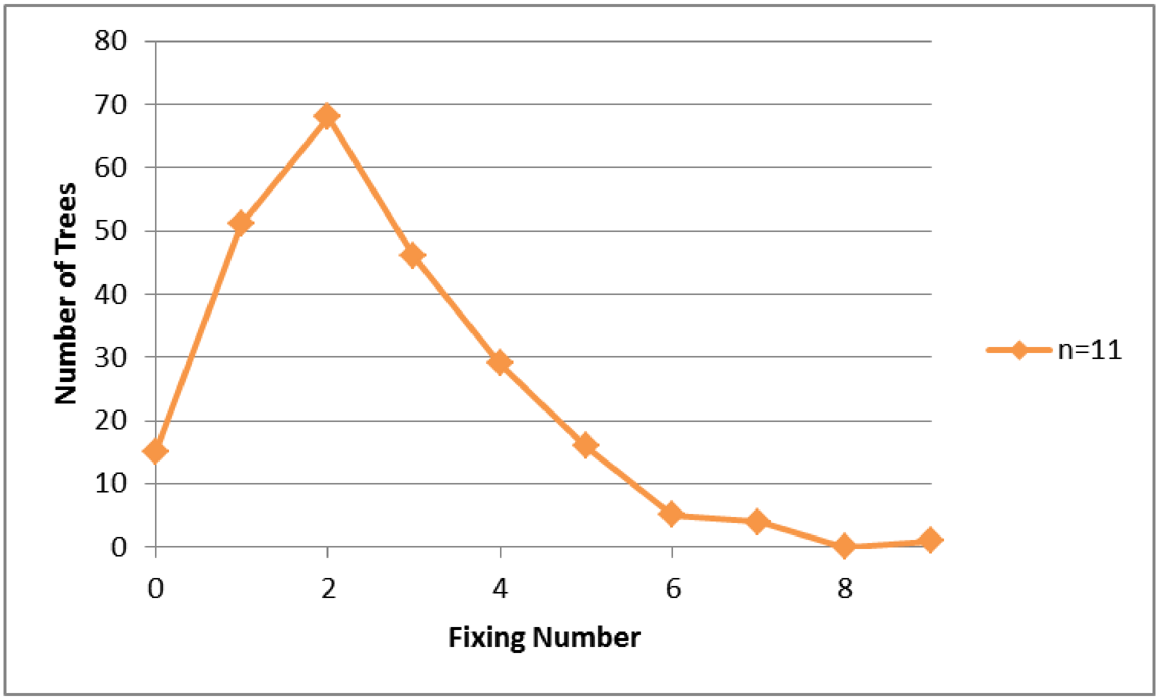


Figure 27 Distribution of Fixing Numbers of Trees of Order 11

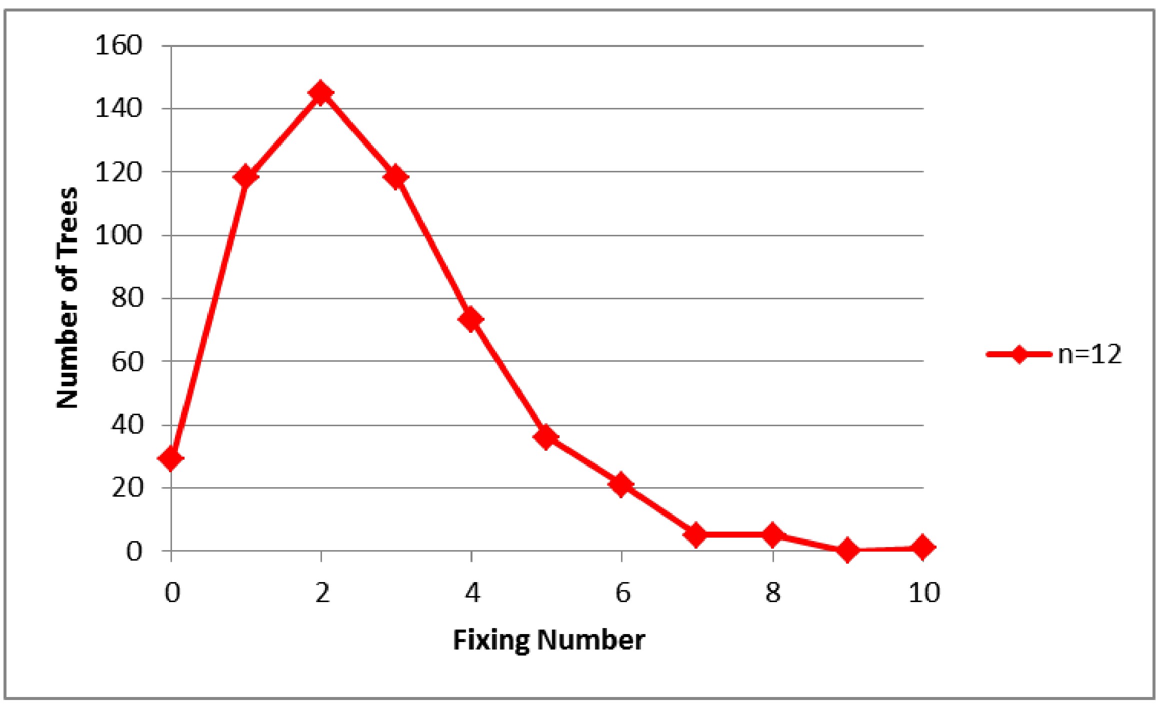


Figure 28 Distribution of Fixing Numbers of Trees of Order 12

The following chart shows the actual percentages of small trees with given fixing numbers. The enumeration of trees of various fixing n umbers from the previous four charts have been normalized so as to illustrate the similarities in the probability distributions. Although this computation has only been done for relatively small trees, a pattern is apparent, and it is the hope of the author that future work will be able to discover a general equation for the distribution of trees on n vertices.

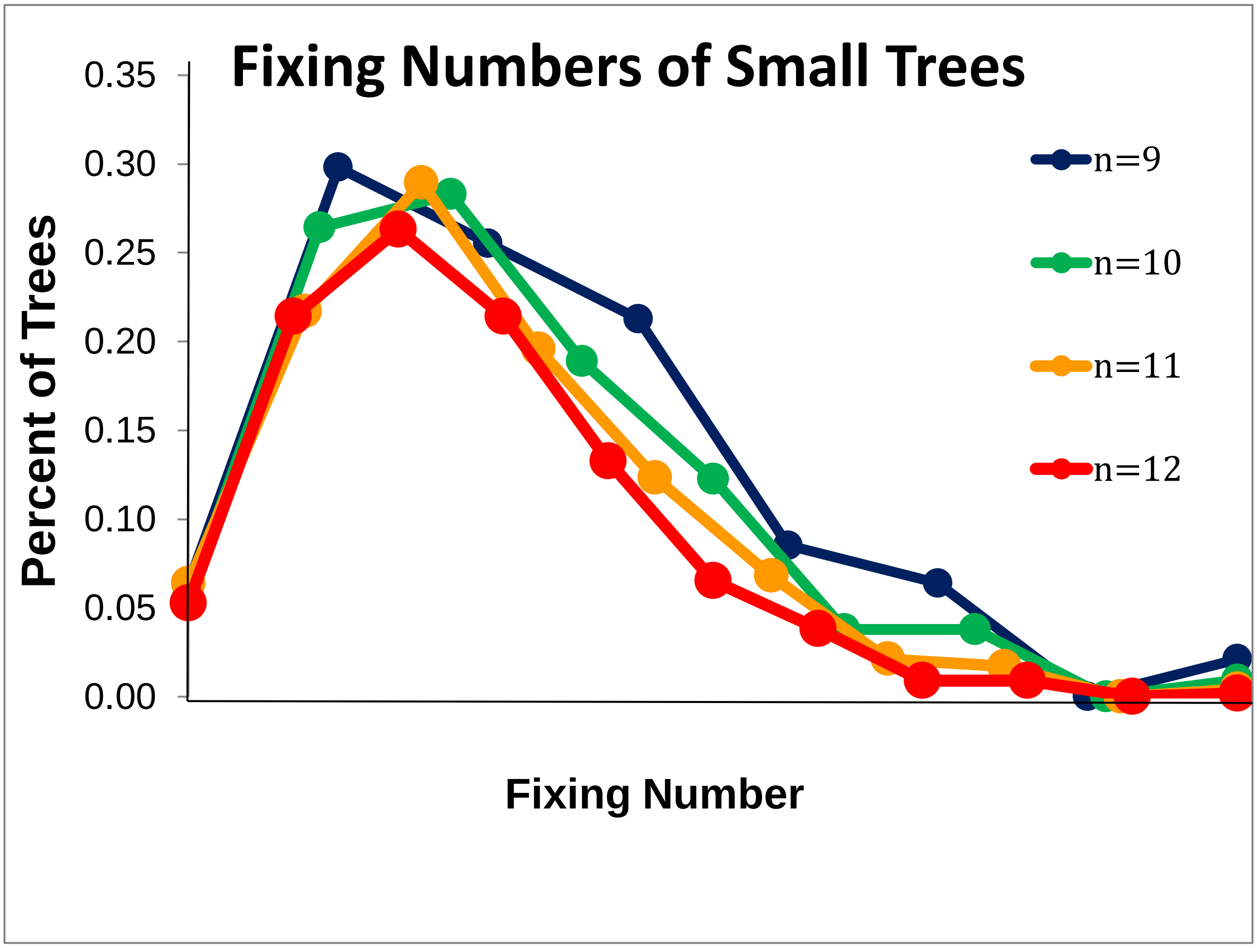
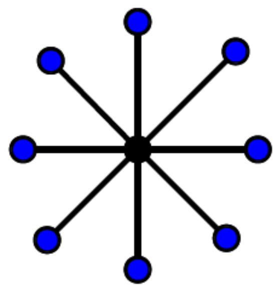
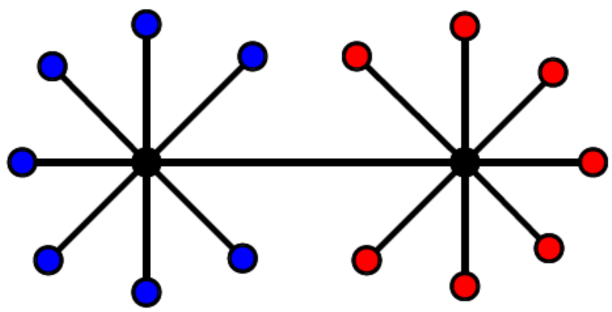
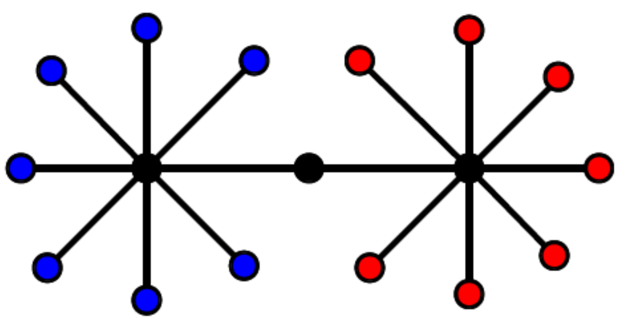
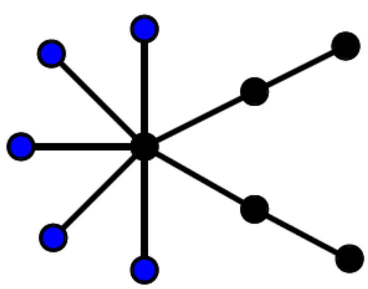
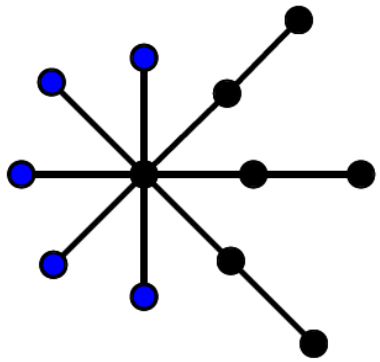
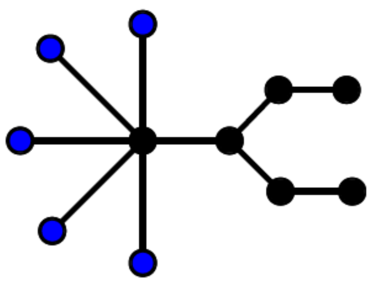
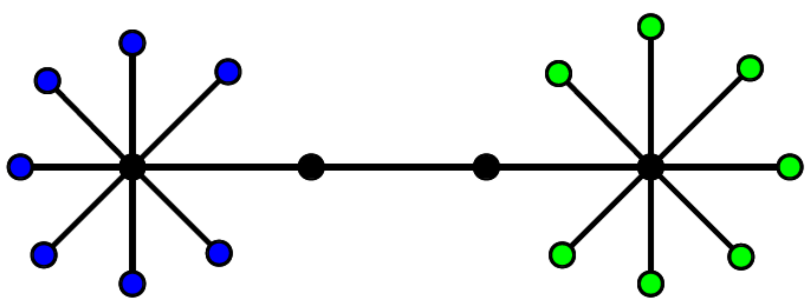


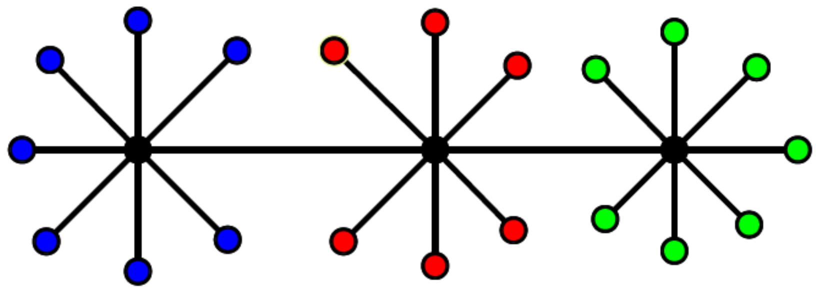
Figure 29 Normalized Distributions of Fixing Numbers of Trees of Order 9 through 12

1.3 Generalization to Larger Trees

As a preliminary step towards discovering the equation for the distribution of trees of order n. The number of trees of order n and fixing number $n - x$ for some small values of x has been calculated using elementary combinatorics and the results can be seen in the following chart. The actual computations are rather tedious but are based on the descriptions provided in the following table.

Fixing Number	**Types of Trees of order n**	**Number of Non-Isomorphic Trees of Order n with Given Fixing Number**
$\boldsymbol{n-2}$	 There are $n-1$ blue vertices.	1
$\boldsymbol{n-3}$		0
$\boldsymbol{n-4}$	 In total, there are $n-2$ red and blue vertices. There is at least 1 vertex of each color.	$\left\lfloor\frac{n-2}{2}\right\rfloor$
$\boldsymbol{n-5}$	 In total, there are $n-3$ red and blue vertices. There is at least 1 vertex of each color  There are $n-5$ blue vertices.	$\left\lfloor\frac{n-3}{2}\right\rfloor+1$

Fixing Number	Types of Trees of order n	Number of Non-Isomorphic Trees of Order n with Given Fixing Number
$n-6$	 There are $n-7$ blue vertices  There are $n-6$ blue vertices.  In total, there are $n-4$ blue and green vertices. There is at least 1 vertex of each color.	$\left\lfloor\frac{n-4}{2}\right\rfloor\left\lceil\frac{n-4}{2}\right\rceil + \left\lfloor\frac{n-4}{2}\right\rfloor + 1$

Fixing Number	Types of Trees of order n	Number of Non-Isomorphic Trees of Order n with Given Fixing Number
	 In total, there are $n-4$ blue, red, and green vertices. There is at least 1 vertex of each color.	

2 Distribution of Fixing Numbers of Homeomorphically Irreducible Trees

A homeomorphically irreducible tree is a tree that does not contain any vertices of degree 2 [19]. These types of trees are also known as series reduced trees and topological trees. They are a very interesting sub-classification of tree to study because the irreducibility requirement introduces a sort of parity in fixing number that will become more apparent when we examine the distribution of the fixing numbers of these trees below.

The following charts show the distribution of fixing numbers of small homeomorphically irreducible trees. There is a noticeable pattern of alternating high and low values. This is conjectured to be due to the occurrence of cherries within the trees and the requirement that each cherry contain two leaf vertices, but there is no conclusive proof available to explain this pattern at the current time.

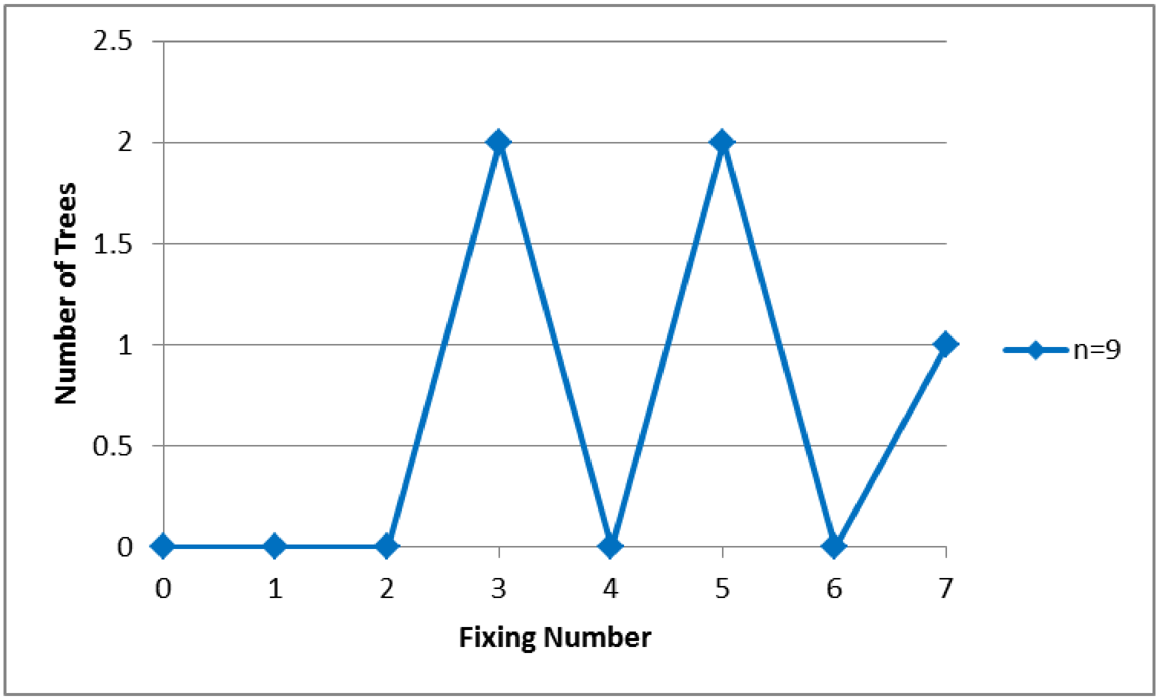


Figure 30 Distribution of Homeomorphically Irreducible Trees of Order 9

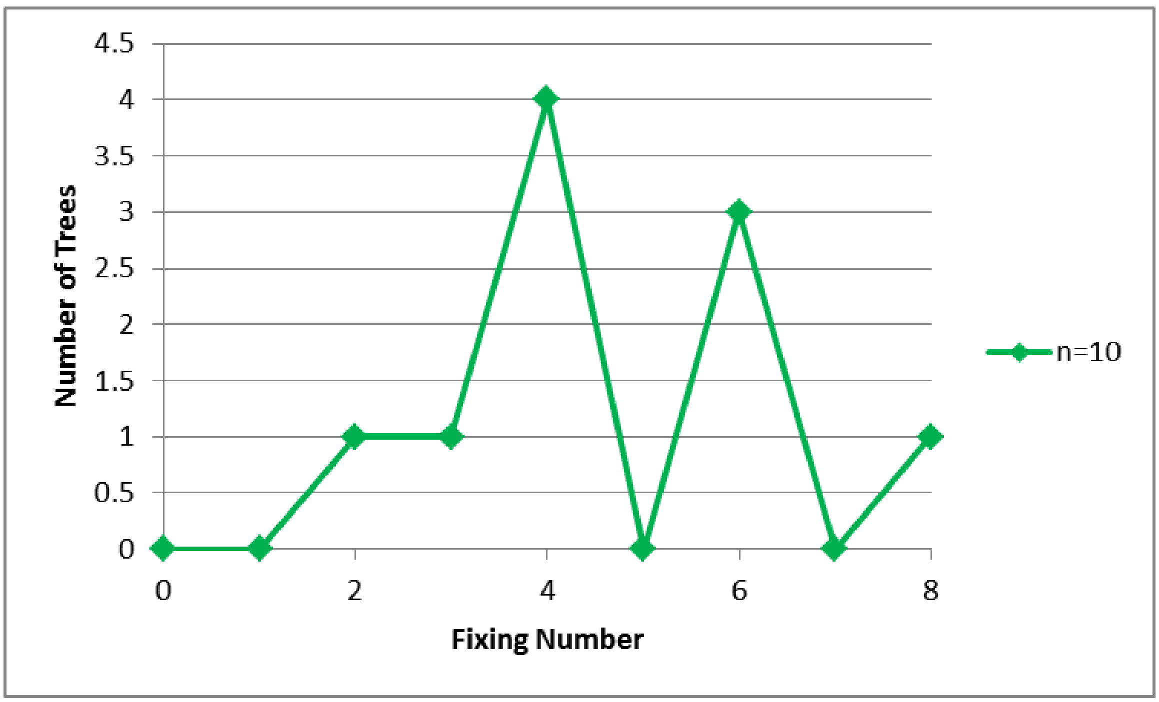


Figure 31 Distribution of Homeomorphically Irreducible Trees of Order 10

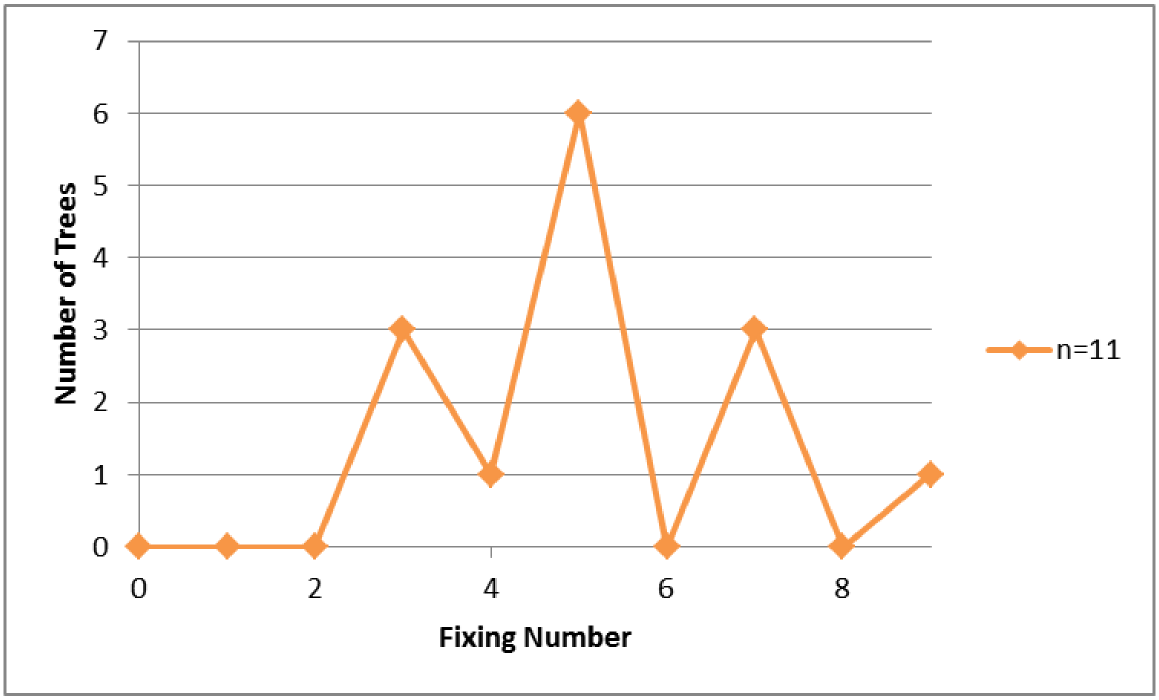


Figure 32 Distribution of Homeomorphically Irreducible Trees of Order 11

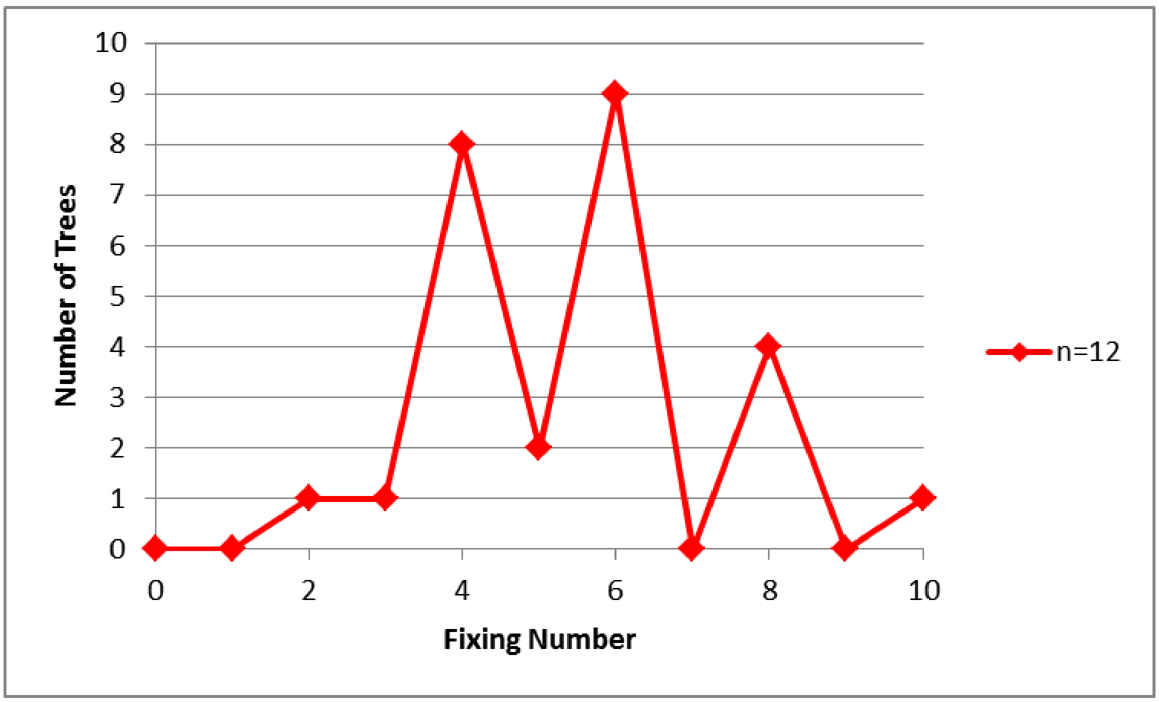


Figure 33 Distribution of Homeomorphically Irreducible Trees of Order 12

Observation 16.

There are no homeomorphically irreducible trees of order 3. This follows since the only tree of order 3 is P_3, which is not homeomorphically irreducible.

■

2.1 Fixing Numbers of Small Homeomorphically Irreducible Trees

A **cherry** is an induced subgraph consisting of 2 monovalent vertices (vertices of degree 1) which are both adjacent to a third vertex. This third vertex may also be adjacent to other vertices [11]. Alternatively, this subgraph is known as a claw [20].

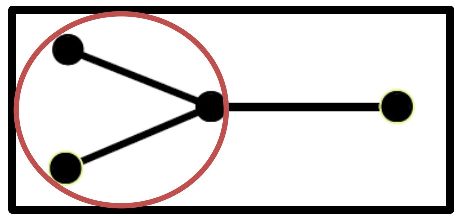


Figure 34 Cherry

Lemma 17.

Every homeomorphically irreducible tree other than P_1 contains one or more cherries.

2.2 Homeomorphically Irreducible Trees with Small Fixing Numbers

Observation 18.

The only homeomorphically irreducible tree with fixing number 0 is P_1.

Proof:

Every homeomorphically irreducible tree with order ≥ 3 must contain a cherry. Any graph with a cherry has fixing number ≥ 1. The only homeomorphically irreducible trees with order < 3 are P_1 and P_2. P_1 is the only one of these with fixing number 0.

■

Observation 19.

The only homeomorphically irreducible tree with fixing number 1 is P_2.

Proof:

Every homeomorphically irreducible tree with order > 3 must contain at least 2 cherries, possibly sharing vertices. This follows directly from the fact that each non-leaf vertex must have degree at least 3. The only homeomorphically irreducible trees with order ≤ 3 are P_1 and P_2. P_2 is the only one of these with fixing number 1.

■

Theorem 20.

If n is even and $n \geq 4$, then there is one homeomorphically irreducible tree of order n with fixing number 2. If n is odd, then there are no homeomorphically irreducible trees of order n with fixing number 2.

Proof:

There are 2 ways in which a homeomorphically irreducible tree can have fixing number 2. The first of these is $K_{1,3}$. This is the only homeomorphically irreducible tree that contains 2 cherries which share vertices (3 monovalent vertices which are all adjacent to a common vertex, because any other homeomorphically irreducible graph that contained such a graph as an induced subgraph would also have to contain at least one additional cherry, giving it a fixing number of at least 3. All other homeomorphically irreducible trees with fixing number 4 must contain exactly 2 cherries which do not have any vertices in common. Since each cherry contains 3 vertices, such a graph must contain at least 6 vertices, with any additional vertices not part of either of the cherries. In order for there not to be any additional cherries, the additional vertices must come in pairs of a leaf vertex and the non-leaf vertex to which that leaf is adjacent. This necessitates an even number of vertices and up to isomorphism permits only one such homeomorphically irreducible tree of order n (for n even).

■

Figure 35 shows the unique homeomorphically irreducible tree of even order with fixing number 2.

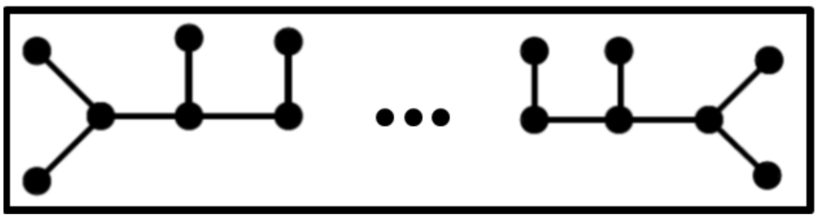
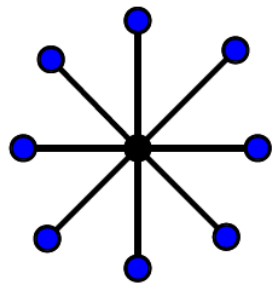
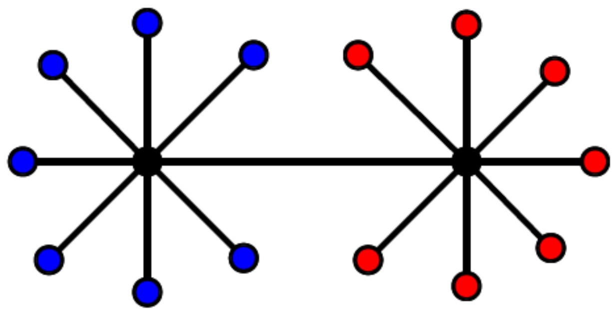
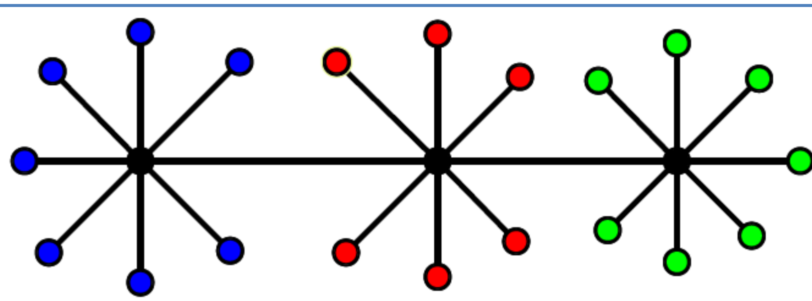


Figure 35 Construction of Homeomorphically Irreducible Trees of Even Order with Fixing Number 2

2.3 Homeomorphically Irreducible Trees with Large Fixing Numbers

The following chart shows the classes of homeomorphically irreducible trees with large fixing numbers and counts the number of homeomorphically irreducible trees of order n with fixing number $n - x$ for small values of x. The total counts displayed in the rightmost column have been obtained through fairly elementary combinatorics, the calculations of which have been omitted.

Fixing Number	**Types of Homeomorphically Irreducible Trees of order n**	**Number of Trees of Order n with Given Fixing Number**
$\boldsymbol{n-2}$	 There are $n-1$ blue vertices.	1
$\boldsymbol{n-3}$		0
$\boldsymbol{n-4}$	 In total, there are $n-2$ red and blue vertices. There is at least 1 vertex of each color.	$\left\lfloor \frac{n-2}{2} \right\rfloor$
$\boldsymbol{n-5}$		0
$\boldsymbol{n-6}$	 In total, there are $n-4$ blue, red, and green vertices. There is at least 1 vertex of each color.	$\left\lfloor \frac{n-4}{2} \right\rfloor \left\lceil \frac{n-4}{2} \right\rceil - 1$

3 Distribution of Fixing Numbers of Strongly Binary Trees

Strongly binary trees are a special class of rooted trees in which the root either has degree 0 or 2. All other vertices either have degree 1 or 3 [21]. As was the case with general rooted trees, the root of a rooted tree will be distinguishable from all other vertices and hence implicitly fixed, but will not contribute to the fixing number of the graph.

Proposition 21.

The only strongly binary tree with fixing number 0 is P_1.

Proof:

All other binary trees will contain one or more cherries, each of which will add one to the fixing number

■

Lemma 22.

All strongly binary trees have an odd number of vertices [21].

Proof:

All binary trees have one vertex of degree 2 (the root) and k vertices of degree 1, and $n-k-1$ vertices of degree 3, where n is the number of vertices. Summing over the degrees of the vertices, we get the following:

$2n-2=2+k+(n-k-1)(3)$

$0=n-2k+1$

$n=2k-1$

Therefore, n is odd.

■

A right comb is a binary tree in which the left child of every vertex is always monovalent. A left tree is a binary tree in which the right child of every vertex is monovalent [22]. We define a comb as referring ambiguously to either a left or right comb.

Theorem 23.

For each odd n, there is exactly 1 strongly binary tree (up to isomorphism) with fixing number 1.

Proof:

A strongly binary tree with fixing number 1, must by definition contain a single cherry and consequently consist of a single vertex of degree 2 (the root), 3 vertices of degree 1, and $n-4$ vertices of degree 3. The only way in which this can be achieved is through the comb structure shown here, of which there is only a single realization up to isomorphism for a given order.

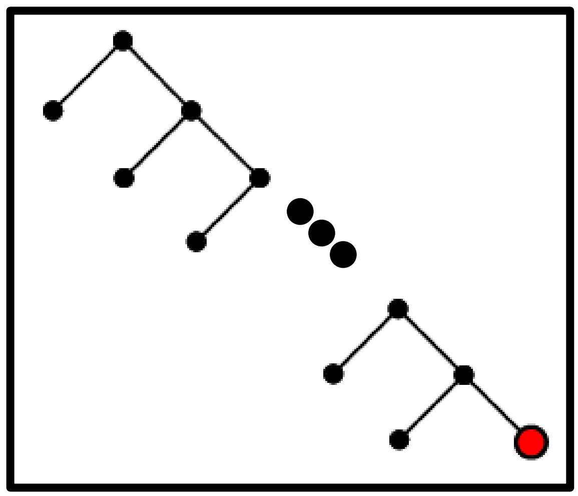


Figure 36 Comb with Fixing Number 1

■

Theorem 24.

(a) For each odd $n, n \equiv 1(mod4)$, there are exactly $\frac{(n-5)(n-1)}{16}$ strongly binary trees of order n (up to isomorphism) with fixing number 2.

(b) For each odd $n, n \equiv 3(mod4)$, there are exactly $\frac{(n-3)^2}{16}$ strongly binary trees of order n (up to isomorphism) with fixing number 2.

Proof:

In order for a binary tree to have fixing number 2, it must contain exactly 2 cherries. Figure 37 shows the general configuration of such a tree. A simple combinatorial argument counts the number of such trees as detailed above.

■

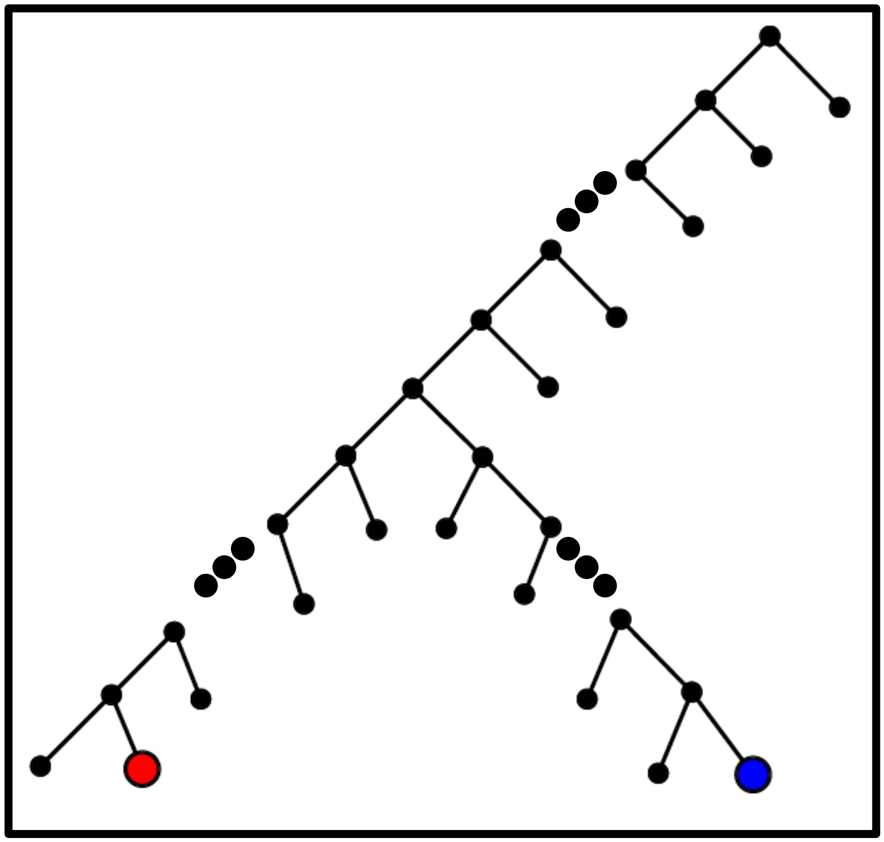


Figure 37 Strongly Binary Tree with Fixing Number 2

Lemma 25.

Every minimum fixing set of a binary tree will consist of 1 vertex from each cherry in that tree.

Proof

By definition, any minimum fixing set must contain exactly one vertex from each cherry. Since we are dealing with binary trees, there are no rigid branches that are not isomorphic to P_2, so fixing one vertex from each cherry will fix the graph.

■

Corollary 26.

Let k be the number of cherries in a binary tree, T. T has 2^k minimum fixing sets.

Theorem 27.

If T is a strongly binary tree on n vertices, then

$$\text{fix}(T) \leq \frac{n+1}{4}$$

Equation 28

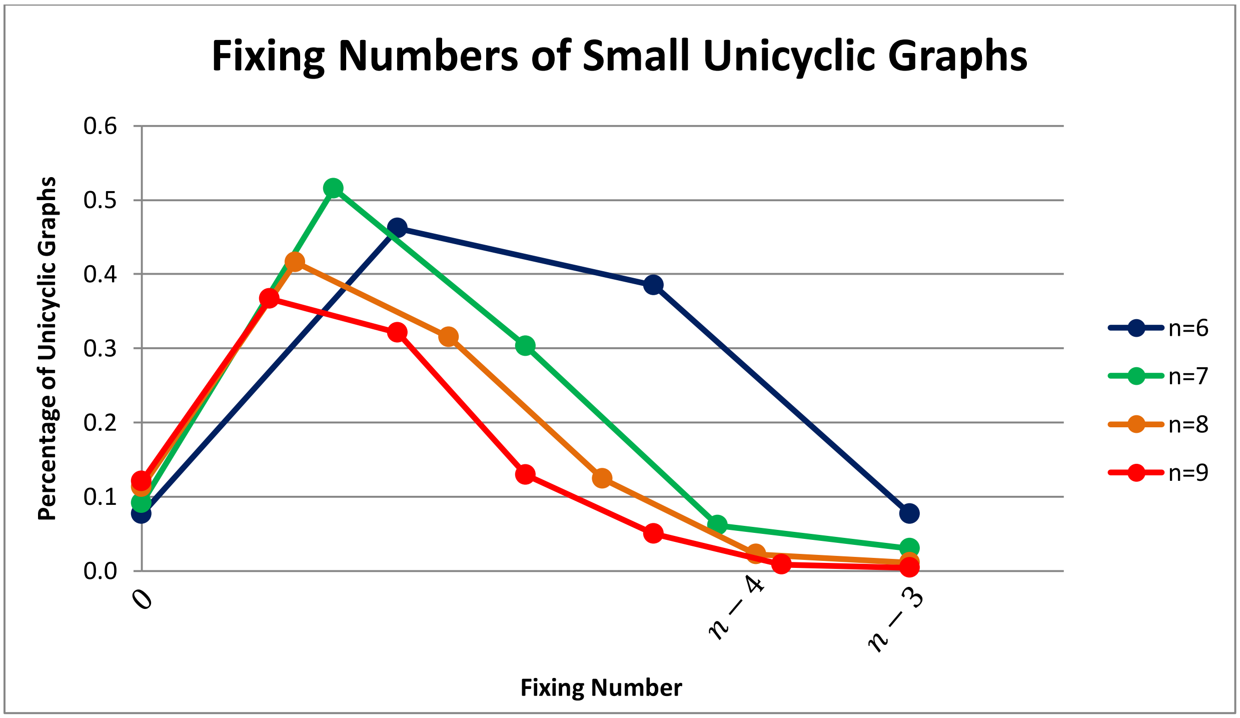
Proof:

Binary trees have at most $\frac{n+1}{2}$ leaf vertices. By the preceding lemma, we know that exactly half of the leaf vertices in a binary tree will be in a minimum fixing set.

■

4 Distribution of Fixing Numbers of Unicyclic Graphs

We now examine the fixing numbers of unicyclic graphs, which are connected graphs containing a single cycle [8]. The following chart shows the distribution of fixing numbers of the small unicyclic graphs.



Theorem 28.

Most unicyclic graphs are not rigid (have fixing number at least 1).

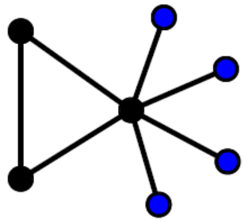
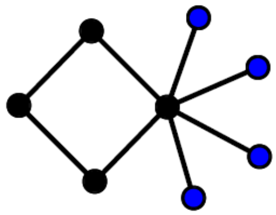
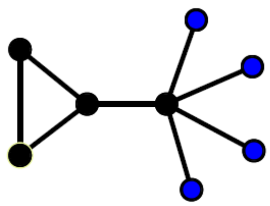
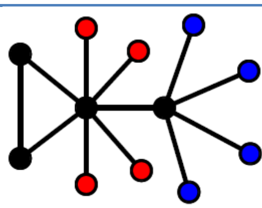
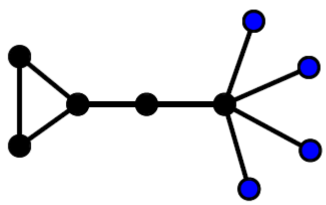
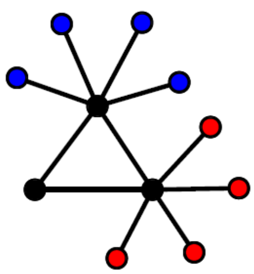
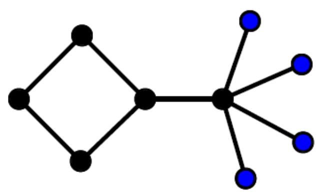
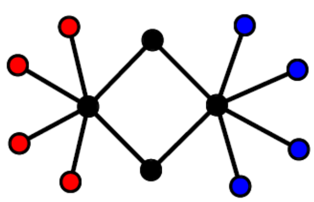
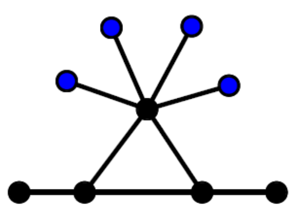
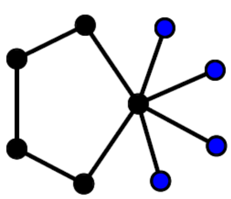
Proof:

Let G be a unicyclic graph of order n. There is a single possible realization of G which is a cycle. All other possible realizations will be composed a cycle of at most order $n-1$ with rooted trees incident with some or all of the vertices of the cycle. Most rooted trees contain one or more cherries, each of which contributes 1 to the fixing number, so consequently, most unicyclic graphs will have fixing number at least 1.

■

4.1 Unicyclic Graphs with Large Fixing Numbers

The right tail of the curves above showing small percentages for unicyclic graphs of order n and fixing number $n - 3$ or $n - 4$ is explained in the chart below.

Fixing Number	Types of Unicyclic Graphs of order n	Number of Unicyclic Graphs of Order n with Given Fixing Number
$\boldsymbol{n-3}$		1
$\boldsymbol{n-4}$	 	2
$\boldsymbol{n-5}$	      	$2n-5$

Conclusions

1 Applications

Currently, fixing numbers have not been applied to any real world problems. However, symmetry is present in many areas ranging from robotics to chemistry, and it is our belief that providing information on a metric to define the level of symmetry has the potential to be used in a wide variety of fields.

2 Future Work

Gibbons and Laison [1] introduced the concept of the fixing set of a group, which is defined to be the set of fixing numbers of all graphs with that group as their automorphism group. At this time, little is known about fixing sets of groups

Bibliography

1. *Fixing Numbers of Graphs and Groups.* **Gibbons, Courtney and Laison, Josh.** March 20, 2009, Electronic Journal of Combinatorics, Vol. 16, p. R39.

2. *Destroying Automorphisms by Fixing Nodes.* **Erwin, David and Harary, Frank.** s.l. : Science Direct, 2006, Discrete Mathematics, pp. 3244-3252.

3. *Identifying Graph Automorphisms Using Determining Sets.* **Boutin, Debra L.** September 7, 2006, Electronic Journal of Combinatorics, Vol. 13, p. R78.

4. **Skiena, S.** *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica.* Reading : Addison-Wesley, 1990. p. 93.

5. **Bollobas, B.** *Modern Graph Theory.* New York : Springer-Verlag, 1998.

6. *Symmetry Breaking in Graphs.* **Albertson, Michael O and Collins, Karen L.** 1, 1996, Electronic Journal of Combinatorics, Vol. 3, p. 17.

7. **Harary, F and Palmer, E M.** *Graphical Enumeration.* New York : Academic Press, 1973. p. 220.

8. **Harary, F.** *Graph Theory.* Reading : Addison-Wesley, 1994.

9. **Gross, J T and Yellen, J.** *Graph Theory and Its Applications.* 2. Boca Raton : CRC Press, 2006.

10. *The Determining Number of a Cartesian Product.* **Boutin, Debra.** 2, New York, NY : John Wiley & Sons, Inc, June 2009, Journal of Graph Theory, Vol. 61, pp. 77-87.

11. *Asymmetric Graphs.* **Erdos, P and Renyi, A.** Budapest : Hungarian Academy of Sciences, 1963, Acta Math. , Vol. 14, pp. 295-315.

12. **Ramirez Alfonsin, Jorge L and Reed, Bruce A.** *Perfect Graphs.* West Sussex : Wiley, 2001.

13. *Dynamic Survey of Graph Labeling.* **Gallian, J.** 14, January 3, 2007, Electronic Journal of Combinatorics. DS6.

14. *On a Problem of Graph Theory.* **Erdös, Paul, Renyi, A. and Sos, V. T.** 1966, Studia Scientiarum Mathematicarum Hungarica I, pp. 215-235.

15. *On the Determining Number and Metric Dimension of Graphs.* **Cáceres, José, et al.** April 19, 2010, Electronic Journal of Combinatorics, Vol. 17, p. R63.

16. **Mowshowitz, Abbe and Mitsou, Valia.** Entropy, Orbits, and Spectra of Graphs. [ed.] Matthias Dehmer and Frank Emmert-Streib. *Analysis of Complex Networks: From Biology to Linguistics.* Weinheim : Wiley-VCH Verlag GmbH & Co., 2009.

17. **de Montmort, P R.** *Essai d'analyse sur les jeux de hasard.* Paris : s.n., 1708.

18. **Steinbach, Peter.** *Field Guide to Simple Graphs.* Albuquerque : s.n., 1990.

19. **Bergeron, F, Leroux, P and Labelle, G.** *Combinatorial Species and Tree-Like Structures.* Cambridge : Cambridge University Press, 1998. pp. 188, 283-284, 291, 337.

20. *On the Uniqueness of the Triangular Association Scheme.* **Hoffman, A J.** 1960, Ann. Math. Stat.

21. **Finch, S R.** *Mathematical Constants.* Cambridge : Cambridge University Press, 2003.

22. *Toward a Language Theoretic Proof of the Four Color Theorem.* **Cooper, Bobbe J, Rowland, Eric S and Zeilberger, Doron.** June 5, 2010, eprint axXiv. Xiv.1006.1324.

23. *Automorphisms and Distinguishing Numbers of Geometric Cliques.* **Albertson, Michael O and Boutin, Debra L.** 4, Secaucus, NJ : Springer-Verlag New York, Inc, June 2008, Discrete and Computational Geometry, Vol. 39, pp. 778-785.

24. *Small Label Classes in 2-Distinguishing Labelings.* **Boutin, Debra L.** November 29, 2008, ARS Mathematica Contemporanea, pp. 154-164.

25. *Using Determining Sets to Distinguish Kneser Graphs.* **Albertson, Michael O and Boutin, Debra L.** January 2009, 2007, Electronic Journal of Combinatorics, Vol. 14, p. R20.

26. *The Distribution of Degrees ina Large Random Tree.* **Robinson, R W and Schwenk, A J.** 12, 1975, Discrete Math.

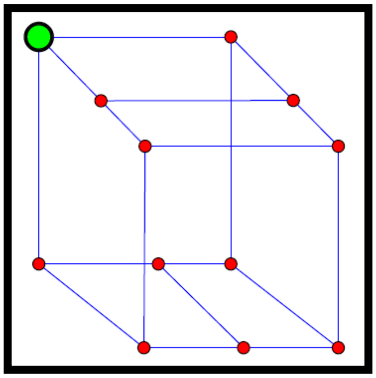
27. **Knuth, D E.** *The Art of Computer Programming.* 3. Reading : Addison-Wesley, 1997. pp. 385-399. Vol. 1.

28. **Polya, G and Read, R C.** *Combinatorial Enumeration of Groups, Graphs, and Chemical Compounds.* s.l. : Springer-Verlag, 1987. p. 63.

Appendices

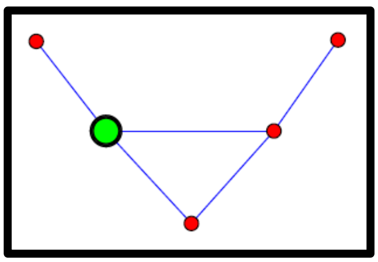
1 Fixing Numbers of Special Graphs

1.1.1 Bidakis Cube



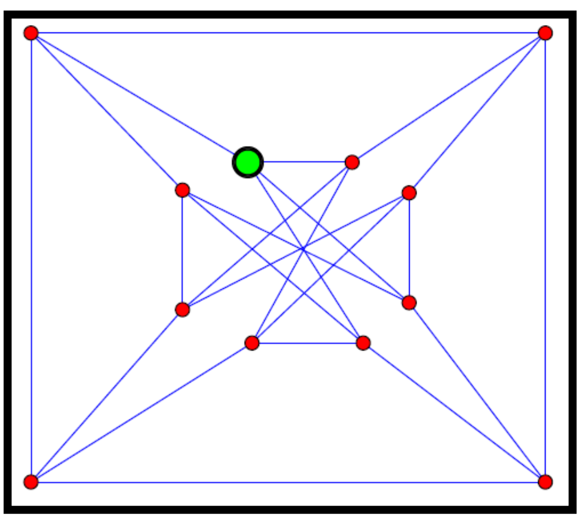
Fixing Number = 1.

1.1.2 Bull Graph



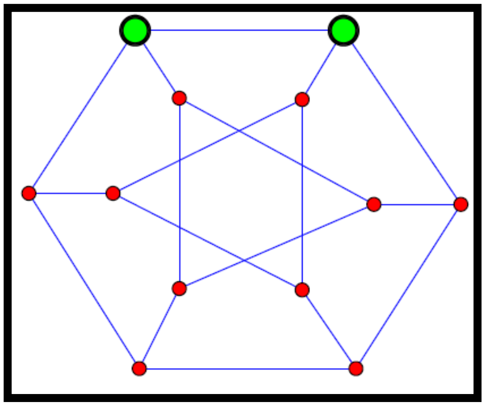
Fixing Number = 1.

1.1.3 Chvatal Graph



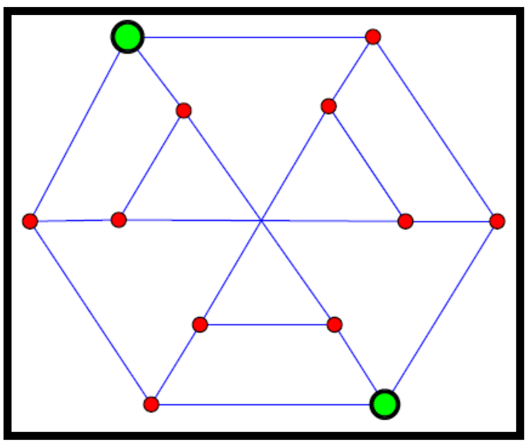
Fixing Number = 1.

1.1.4 Durer Graph



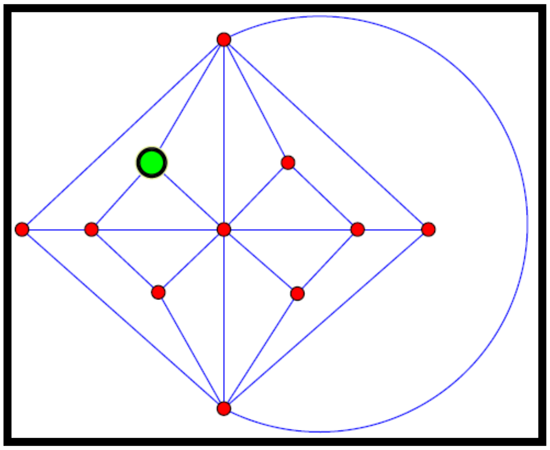
Fixing Number = 2.

1.1.5 Franklin Graph



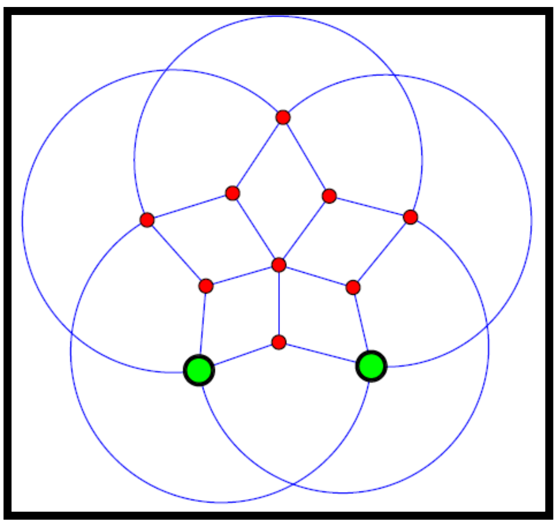
Fixing Number = 2

1.1.6 Goldner-Harary Graph



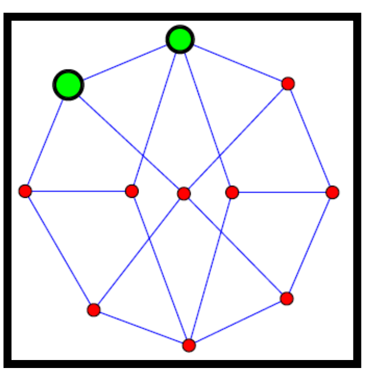
Fixing Number = 1.

1.1.7 Groetzsch Graph



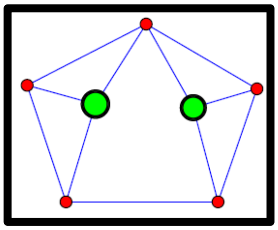
Fixing Number = 2.

1.1.8 Herschel Graph



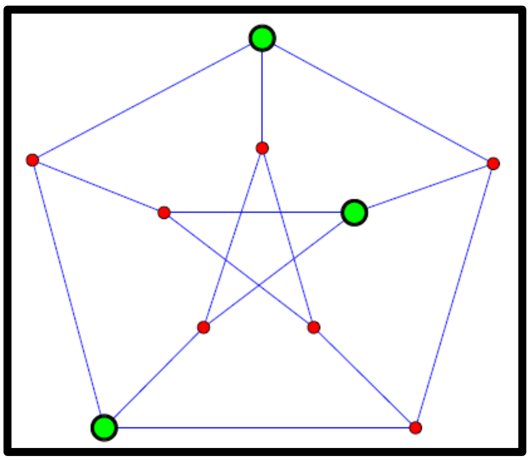
Fixing Number = 2.

1.1.9 Moser Spindle



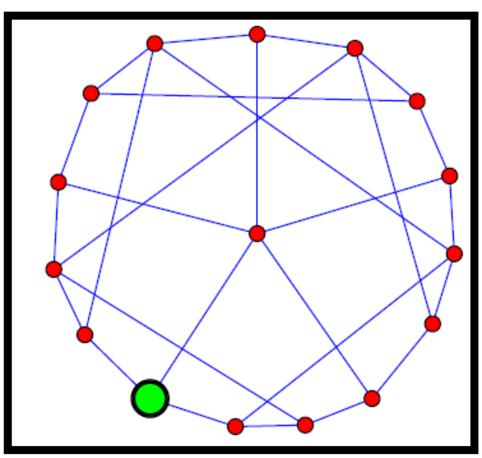
Fixing Number = 2.

1.1.10 Peterson Graph



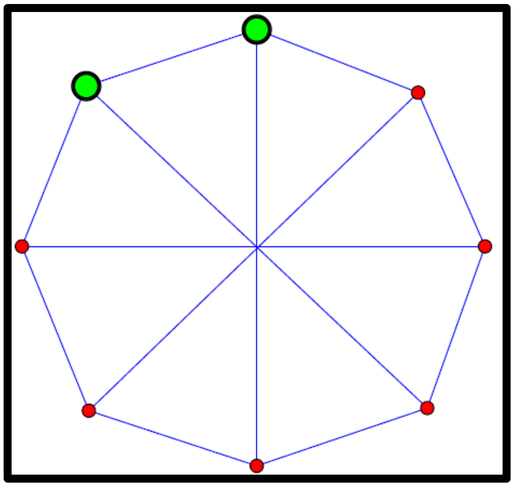
Fixing Number = 3.

1.1.11 Sousselier Graph



Fixing Number = 1.

1.1.12 Wagner Graph



Fixing Number = 2.