MINIMAL RESTRAINED DOMINATION ALGORITHMS ON TREES USING DYNAMIC PROGRAMMING

A Thesis
by
JEREMY BOOKER

Submitted to the Graduate School
at Appalachian State University
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

December 2013
Department of Computer Science

MINIMAL RESTRAINED DOMINATION ALGORITHMS ON TREES USING DYNAMIC PROGRAMMING

A Thesis
by
JEREMY BOOKER
December 2013

APPROVED BY:

Alice A. McRae
Chairperson, Thesis Committee

Dolores A. Parks
Member, Thesis Committee

James Wilkes
Member, Thesis Committee

James Wilkes
Chairperson, Department of Computer Science

Edelma D. Huntley
Dean, Cratis Williams Graduate School

Abstract

MINIMAL RESTRAINED DOMINATION ALGORITHMS ON TREES USING DYNAMIC PROGRAMMING

Jeremy Booker
B.S., Appalachian State University
M.S., Appalachian State University

Chairperson: Alice A. McRae

In this paper we study a special case of graph domination, namely minimal restrained dominating sets on trees. A set $S \subseteq V$ is a dominating set if for every vertex $u \in V - S$, there exists $v \in S$ such that $uv \in E$. A set $S \subseteq V$ is a restrained dominating set if every vertex in $V - S$ is adjacent to a vertex in S and another vertex in $V - S$. A restrained dominating set S is a minimal restrained dominating set if no proper subset of S is also a restrained dominating set. We give a dynamic programming style algorithm for generating largest minimal restrained dominating sets for trees and show that the decision problem for minimal restrained dominating sets is NP-complete for general graphs. We also consider independent restrained domination on trees and its associated decision problem for general graphs.

Acknowledgments

First, I wish to thank my thesis advisor, Dr. Alice McRae. Her excellent teaching, advice, and encouragement throughout my college career have been invaluable. I would also like to thank the members of the thesis committee, Dr. Dolores Parks and Dr. James Wilkes, for their service and inspiration in my early studies.

In addition, I wish to thank Mr. Brian Brown, Director of Electronic Student Services at Appalachian State University, for his support an encouragement in this pursuit. I am deeply indebted for your kind mentorship.

Finally, I would like to thank my parents, who encouraged my curiosity in technology from a very young age and have supported me since the beginning of my studies.

Dedication

This thesis is dedicated to my brilliant, loving, and always supportive wife, Kristi.

Table of Contents

List of Tables

List of Figures

Foreword

This thesis is intended to be the basis of a technical publication. Therefore, this thesis is formatted according to the guidelines of the Association of Computing Machinery (ACM). The style guidelines of the Cratis D. Williams Graduate School at Appalachian State University are substituted as necessary according to the Graduate School requirements.

Chapter 1: Introduction

1.1 Purpose and History

Let G = (V, E) be a graph. A set $S \subseteq V$ is a dominating set in a graph if for every vertex $u \in V - S$, there exists $v \in S$ such that $uv \in E$. In other words, a set is a dominating set if every vertex not in the set is adjacent to a vertex in the set. Dominating sets have been studied extensively in graph theory. In 1990, Hedetniemi and Laskar published a bibliography of graph domination with more than 300 citations [8]. By 1998, the textbook Fundamentals of Domination in Graphs listed a bibliography with more than 1000 entries [7]. In 2006, Hedetniemi published a list of several hundred graph domination problems that remained open. One of the problems listed was minimal restrained domination [6]. This thesis addresses this open problem.

In graph domination theory, the notation $\gamma(G)$ is used to denote the cardinality of a minimum dominating set in a graph G. The most studied variation of the domination number is the independent domination number. In an independent dominating set, no two vertices in the dominating set can be adjacent. The cardinalities of a minimum and maximum independent dominating set in a graph are denoted by $i(G)$ and $\beta(G)$, respectively. The famous eight queen's chessboard problem is an example of maximum independent dominating problem.

There also is a maximum version of the dominating set problem. The entire set of vertices V in a graph is a dominating set, by the definition of dominating set, so the maximum domination problem is not very interesting. Instead, what is studied is the maximum minimal dominating set problem—the problem of finding a largest dominating set in which no proper subset is also a dominating set. Each vertex v in a minimal dominating set must have a purpose;

there must be at least one vertex w that would not be dominated if v were removed from the set. If a vertex w is dominated by only one vertex v in a set S, then w is called a private neighbor of v. A vertex v that is in S, but has no other neighbors in S, is said to be its own private neighbor. A dominating set S is minimal if every vertex in S has at least one private neighbor.

A set $S \subseteq V$ is a restrained dominating set if every vertex in $V - S$ is adjacent to a vertex in S and another vertex in $V - S$. The notion of a restrained dominating set was first defined by Telle [9] in 1994 and then expanded further by Domke, Hattingh, Hedetniemi, Laskar and Markus [2]. The restrained domination number of a graph G, denoted $\gamma_r(G)$, is the cardinality of a minimum restrained dominating set. The works of Telle and Domke et. al. focused on this graph parameter. A restrained dominating set S is a minimal restrained dominating set if no proper subset of S is also a restrained dominating set. The minimal restrained domination number of a graph G, denoted $\Gamma_r(G)$, is the cardinality of the largest minimal restrained dominating set.

The decision problem for the minimum restrained domination number of a graph has been shown to be NP-complete for bipartite and chordal graphs [2]. A linear-complexity algorithm for finding the minimum restrained domination number of a tree is given in [2]. For general graphs and trees, however, the minimal restrained domination number is an open problem [6]. In [6], Hedetniemi challenged readers with the problems of finding a linear algorithm for trees for minimal restrained domination and an NP-completeness proof for minimal restrained domination in general graphs.

Using dynamic programming techniques, this thesis provides a linear algorithm for finding the largest minimal restrained dominating set for trees. It also shows that minimal restrained dominating set is NP-complete for general graphs.

Some variations of NP-hard domination problems have polynomial solutions when there is an additional requirement that the dominating set is also an independent set. We did not find any references to independent restrained domination in the literature, and this problem was examined first, since it is potentially easier. This thesis first explores the notion of an independent restrained dominating set in a graph. We will show that not all graphs have independent restrained dominating sets, and even the problem of determining if a graph has an restrained independent dominating set is NP-complete. We also provide a linear algorithm for determining if a tree has an independent restrained dominating set.

1.2 Applications of Restrained Domination

A possible application for minimal restrained dominating sets is any situation in which one group needs to supervise a subordinate group using the minimal number of supervisors possible, but at the same time ensure the supervisors are held accountable by never allowing a supervisor to be alone with subordinate. Hattingh gives an example of this relationship in terms of guards and prisoners [5]. The vertices in S are the guards, while the vertices in $V - S$ are the prisoners. In this way, a guard can supervise every prisoner, but every prisoner is also in view of another prisoner.

A possible application for independent restrained dominating sets is the location of product distribution centers or hospitals where a certain level of redundancy is desired. In this case, vertices could represent cities. Vertices in S represent cities with a distribution center and edges represent transportation routes between cities. Selecting the cities in which to place distribution centers using an independent restrained dominating set guarantees that every city without a distribution center is at least next to a city with one. It also guarantees that every city without a distribution center has a neighbor that also lacks a distribution center. In case of shortages at one distribution center, every city has access to a different center by going through one of its neighbors.

Chapter 2: Background

In this chapter, common graph theory terms used throughout this thesis are defined.

2.1 Graph Definitions

Definition 2.1: A **graph** G is a set V of vertices, and a set E of edges.

Definition 2.2: The **degree** of a vertex v is the number of edges connected to the vertex.

Definition 2.3: A **subgraph** of graph G is any graph H where $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$.

Definition 2.4: A **tree**, T, is a connected acyclic graph.

Definition 2.5: A **leaf** is a vertex v in a tree graph where v has degree equal to 1.

Definition 2.6: The **open neighborhood of a vertex** v in a graph G, denoted as $N(v)$, is the set of all vertices adjacent to v.

Definition 2.7: The **closed neighborhood of a vertex** v in a graph G, denoted as $N[v]$, is the set of vertices $v \cup N(v)$.

Definition 2.8: The **closed neighborhood of a set** of vertices S in a graph G, denoted $N[S]$, is the union of the set S and the vertices in $N[v]$ $\forall v: v \in S$.

Definition 2.9: For a graph $G=(V, E)$ and a set $S \subseteq V$, a vertex w is a **private neighbor** of a vertex v if $N[w] \cap S = \{v\}$.

Definition 2.10: A graph G is a **bipartite graph** if it is possible to partition $V(G)$ into two subsets, V_1 and V_2, such that every element of E(G) joins a vertex of V_1 to a vertex of V_2 [1].

Definition 2.11: An **independent set** on a graph G is a set of vertices S such that no vertex in S is adjacent to another vertex also in S.

Definition 2.12: A set $S \subseteq V$ is a **dominating set** if for every vertex $u \in V - S$, there exists $v \in S$ such that an edge $uv \in E$.

Definition 2.13: The minimum cardinality of a dominating set on a graph G is the **domination number** of G and is denoted by $\gamma(G)$.

Definition 2.14: A **minimal dominating set** on a graph G is a set of vertices S such that no proper subset of S is also a dominating set.

Definition 2.15: The maximum cardinality of a minimal dominating set on a graph G is the **upper domination number** and is denoted by $\Gamma(G)$.

Definition 2.16: A set $S \subseteq V$ is a **restrained dominating set** if every vertex in $V - S$ is adjacent to a vertex in S and another vertex in $V - S$.

Definition 2.17: A **minimal restrained dominating set** S on a graph G is a restrained dominating set S have the property that no proper subset of S is also a restrained dominating set.

Chapter 3: Independent Restrained Dominating Set

We first will use the independent restrained dominating set problem as an introduction to the methods used in Chapter 4. We will consider a dynamic programming technique to create an algorithm for the independent restrained dominating set on trees. We then offer a proof of the problem's NP-complete status for general graphs.

3.1 Definition and characterization

An independent restrained dominating set is an independent set S such that $\forall v \in V - S: N(V) \cap S \neq \emptyset$ and $N(V) \cap (V - S) \neq \emptyset$. In other words, no two vertices in S are adjacent, and every vertex not in S must have at least one neighbor in S and at least one neighbor not in S. Several observations can be made.

Lemma 3.1 *All vertices with deg(v) = 1 must be in S.*

Proof.

Every vertex where *deg(v)* = 1 must be a member of the restrained dominating set because, having only one neighbor, it cannot have both a neighbor in S and a neighbor in $V - S$.

Lemma 3.2 *Not all graphs have an independent restrained dominating set.*

Proof.

Not all graphs will contain an independent restrained dominating set. For example, there is no way to form an independent restrained dominating set on a path on three vertices because the vertices on each end of the path have only one neighbor and must therefore be in S. This leaves the middle vertex without a neighbor in $V - S$.

3.2 Dynamic Programming Algorithm for Trees

Our algorithm uses a dynamic programming method to find independent restrained dominating sets for trees. Dynamic programming algorithms break a problem into simpler sub-problems and then combine the results to solve the larger problem.

We apply this method to restrained dominating sets on trees by breaking the problem into a finite set of trees corresponding to different types of independent restrained dominating set solutions, which represent the base-cases for all trees. We call the various solution types "classes." When the algorithm begins, all vertices are considered as singleton vertices in a forest of sub-trees. For each vertex, we also store a corresponding vector holding the cardinalities for each type of possible solution set. We then take a bottom-up approach to build larger sub-trees by composing two smaller sub-trees and their subsets into a single, larger tree. Two properties of the minimal restrained dominating set problem allow us to apply this method to tree graphs. First, the problem has an optimal substructure. That is, the problem can be recursively reduced to smaller problems, which can be solved more easily. Second, once the problem is reduced in this manner, we can arrive a finite set of base cases, which can then be continually reused. This methodology was first described by Wimer [10].

We define the composition of two tree-subset pairs as follows. Let (T, r, S) represent a tree $T = (v, e)$ rooted at *r* with a set $S \subseteq V(T)$. The composition of two tuples, denoted $(T_1,\ r_1,\ S_1) \circ (T_2,\ r_2,\ S_2) = ((V_1 \cup V_2, E_1 \cup E_2 \cup \{r_1, r_2\}),\ r_1,\ S_1 \cup S_2)$. In other words, the roots of the two trees are joined by an edge from *r1* and *r2* and the corresponding sets are unioned with the root r_1 as the new root. An illustration of an example composition is given in Figure 1. Each vertex that is a member of a set is shaded black.

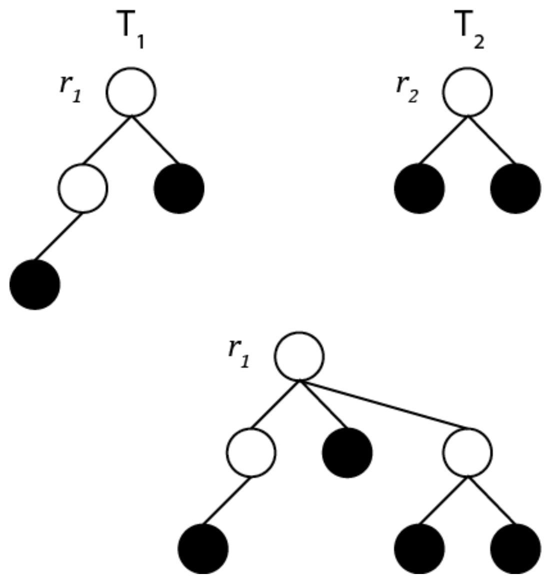

Figure 1. Example composition of subgraphs

The algorithm begins with a rooted tree and compositions are performed from the leaves to the root. A node is composed with its parent only when all of its subtree has been completed. In order to construct our dynamic programming algorithm, we must characterize all of the possible base-cases of tree and subset pairs. For this algorithm, there are five classes of tree and set pairs, which are illustrated in Figure 2. Each vertex that is a member of the subset is shaded black.

Class 1 describes sets with the root in S. The other classes deal with roots in $V - S$. In these cases, we are interested in making sure that eventually the vertex has a neighbor in S and a neighbor in $V - S$. In class 2, the root has at least one neighbor in S but none in $V - S$. Class 3 describes sets where the root has at least one neighbor in $V - S$ but none in S. In class 4, the root has neighbors in both S and $V - S$. Finally, in class 5, the root has no neighbors. It is important to note that each base graph describes an entire class of graphs. Any solution from one of these classes could be replaced by another solution from the same class of graph.

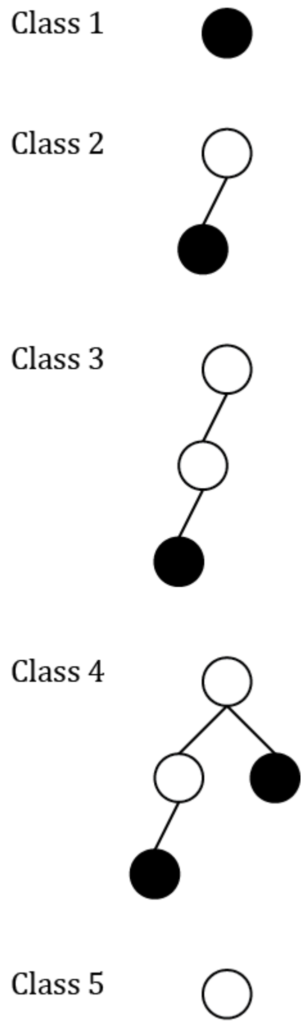

Figure 2. Base subgraphs for Independent Restrained Dominating Set

Next, we consider the effects of composing a graph from each class with another graph of each class. We must determine if the two graphs and their corresponding sets can be composed, and if so, to which class the resulting graph belongs. Table 1 gives the results of composing each of the base classes with each other for the five classes given in Figure 2. In this table, an 'X' is used to denote a composition which was not possible because the resulting tree-subset pair did not form an independent restrained dominating set.

Table 1. Class composition for Independent Restrained Dominating Set

	1	**2**	**3**	**4**	**5**
1	X	X	1	1	X
2	2	4	X	4	X
3	4	3	X	3	X
4	4	4	X	4	X
5	2	3	X	3	X

To demonstrate the correctness of the compositions table, we prove a lemma asserting each composition is correct.

Lemma 3.3 *A class one subgraph cannot be composed with a class one child subgraph, denoted* $[1] \circ [1] = X$.

Proof.

Since both roots are in S, the resulting composition is not be independent. Note the entry in Table 1 is an 'X' for this composition.

Lemma 3.4 *Class five subgraphs cannot be composed as a child subgraph.*

Proof.

Since this leaves the class five subgraph as a leaf vertex with *deg(v)* = 1 in $V - S$, it either will not be dominated or it will not have a neighbor in $V - S$. See Lemma 3.1.

Lemma 3.5 *Class three subgraphs can only be composed as a child of a class one parent graph.*

Proof.

Since the root vertex of a class three subgraph is not dominated, it can only be composed with a class where the root node is an element of S. The only class that meets this criterion is class one. Composing a class three subgraph as a child of any other class will result in the root of the class three graph not being dominated.

Lemma 3.6 $[1] \circ [2]$ *is not a restrained dominating set.*

Proof.

When composed with a class one subgraph, the root of a class two subgraph does not have a neighbor in $V - S$.

Lemma 3.7 $[1] \circ [4] = [1]$

Proof.

The root vertex of a class one subgraph is an element of S, so when it is composed as a parent with a class four subgraph, the root vertex of the resulting composition is still an element of S. All vertices are dominated and are independent, and every vertex in $V - S$ has a neighbor in $V - S$.

Lemma 3.8 $[2] \circ [1] = [2]$

Proof.

The root vertex of a class two subgraph is not an element of S, so when it is composed as a parent with a class one subgraph, the root vertex of the resulting composition still is not an element of S. All vertices are dominated and are independent, and every vertex in $V - S$ has a neighbor in $V - S$.

Lemma 3.9 $[3] \circ [1] = [4]$ *and* $[4] \circ [1] = [4]$

Proof.

A class three subgraph composed with a class one subgraph forms a class four subgraph, which can be seen by inspecting the prototypes in Figure 2. A class four parent subgraph composed with a class one subgraph has the effect of only adding another dominating vertex to the root vertex. All vertices are dominated and are independent, and every vertex in $V - S$ has a neighbor in $V - S$.

Lemma 3.10 $[5] \circ [1] = [2]$

Proof.

This composition can be seen by inspecting the prototypes in Figure 2.

Lemma 3.11 $[2] \circ [2] = [4]$

Proof.

This composition can bee seen by inspecting the prototypes in Figure 2. All vertices are dominated and are independent, and every vertex in $V - S$ has a neighbor in $V - S$.

Lemma 3.12 $[4] \circ [2] = [4]$

Proof.

The root of the parent vertex is an element of $V - S$, so the root of the resulting composition is also an element of $V - S$. The root of the resulting composition is still dominated by a child vertex.

Lemma 3.13 $[3] \circ [2] = [3]$ *and* $[5] \circ [2] = [3]$

Proof.

The composition of $[3] \circ [2] = [3]$ adds additional child vertices, but no child vertex dominates the root vertex. The composition of $[5] \circ [2]$ forms a prototypical class three subgraph, as seen in Figure 2.

Lemma 3.14 $[2] \circ [4] = [4]$

Proof.

The child vertex from the class two parent subgraph dominates the root vertex, while the class four subgraph gives the root a neighbor in $V - S$, which qualifies the resulting composition as a class four subgraph. All vertices are dominated and are independent, and every vertex in $V - S$ has a neighbor in $V - S$.

Lemma 3.15 $[3] \circ [4] = [3]$

Proof.

The root of the resulting subgraph is not dominated, but the root does have neighbors in $V - S$, which qualifies the resulting subgraph as a class three subgraph. All vertices are dominated and are independent, and every vertex in $V - S$ has a neighbor in $V - S$.

Lemma 3.16 $[4] \circ [4] = [4]$

Proof.

The root node of the resulting subgraph is dominated and the root has neighbors in $V - S$. All vertices are dominated and are independent, and every vertex in $V - S$ has a neighbor in $V - S$.

Lemma 3.17 $[5] \circ [4] = [3]$

Poof.

The root node of the resulting subgraph is not dominated and the root has neighbors in $V - S$. All vertices are dominated and are independent, and every vertex in $V - S$ has a neighbor in $V - S$.

The last step in designing the algorithm is to define an initial vector of cardinalities and determine the valid final classes. We must determine the classes of solutions possible for a single vertex. A singleton vertex v could be an element of S, which is a class one solution, or if v is not an element of S then it would be a class five solution. Thus, the initial vector is [1, -, -, -, 0], where '-' means undefined. Classes 2, 3, and 4 sets cannot be formed from a single vertex.

The input to the algorithm is an array of parent vertices *parent*[1…*p*] for the input tree, and the output is the cardinality of the largest independent restrained dominating set. Before the algorithm begins, every vertex is associated with the initial vector given above. The algorithm can then be defined as follows:

Algorithm: Independent Restrained Dominating Set

```
for i=p to n do
begin;
   vector[i] = [1, -, -, -, 0];
end;
for i=p downto 2 do
begin;
j = parent[i];
   vector[j,1] = max(vector[j,1]+vector[i,3], vector[j,1]+vector[i,4]);
   vector[j,2] = max(vector[j,2]+vector[i,1], vector[j,5]+vector[i,1]);
   vector[j,3] = max(vector[j,3]+vector[i,2], vector[j,3]+vector[i,4],
                 vector[j,5]+vector[i,2], vector[j,5]+vector[i,4]);
   vector[j,4] = max(vector[j,2]+vector[i,2], vector[j,2]+vector[i,4],
                 vector[j,3]+vector[i,1], vector[j,4]+vector[i,1],
                 vector[j,4]+vector[i,2], vector[j,4]+vector[i,4]);
   vector[j,5] = '-';
end;
if(vector[1,1] is undefined AND vector[1,4] is undefined)
   return 'no independent restrained dominating set';
endif;
return max(vector[1,1], vector[1,4]);
```

Note, the root vertex of the final parent tree must be handled differently. Some solution classes are allowed for sub-trees because they represent partial solutions that may be completed by a later composition; however, the final solution set for the root must be a complete solution. Only classes one and four represent complete solutions.

The algorithm above finds a maximum cardinality independent restrained dominating set. By selecting the minimum cardinality of each composition choice, the algorithm could find a minimum cardinality independent restrained dominating set.

3.3 NP-Completeness of Independent Restrained Dominating Set

Independent Restrained Dominating Set

Instance: A graph $G = (V, E)$

Question: Does G have an independent restrained dominating set?

Theorem 3.1. *Independent Restrained Dominating Set is NP-complete.*

Proof.

Independent Restrained Dominating Set is in the class NP. A witness for Independent Restrained Dominating Set is a set S of vertices on a graph G. We can verify that S is an independent restrained dominating set in polynomial time by verifying that every vertex in $V - S$ has at least one neighbor in S and at least one neighbor in $V(G) - S$, and that S is independent. Our transformation is from the well-known NP-complete Satisfiability (SAT) problem [3].

Satisfiability

Instance: A set U of variables and a collection C of clauses over U.

Question: Does there exist a satisfying truth assignment for C ?

Given an instance of SAT with a set U of variables and a set C of clauses, generate a corresponding graph with components as follows:

Clause Components:

$\forall c_j \in C$ construct a clause component such that:

$V(c_j) = \{ c_{j1}, c_{j2}, c_{j3})$

$E(c_j) = \{ (c_{j1}, c_{j2}), (c_{j2}, c_{j3})\}$

Variable Components:

$\forall u_j \in U$ construct a variable component such that:

$V(U_i) = \{u_i, \overline{u_i}, x_i\}$

$E(U_i) = \{\{u_i, \overline{u_i}\}, \{u_i, x_i\}, \{\overline{u_i}, x_i\}\}$

Communication Edges:

For each clause c_j, add edges from c_{j1} to the literal vertices in the variable components that correspond to the literals in the clause c_j.

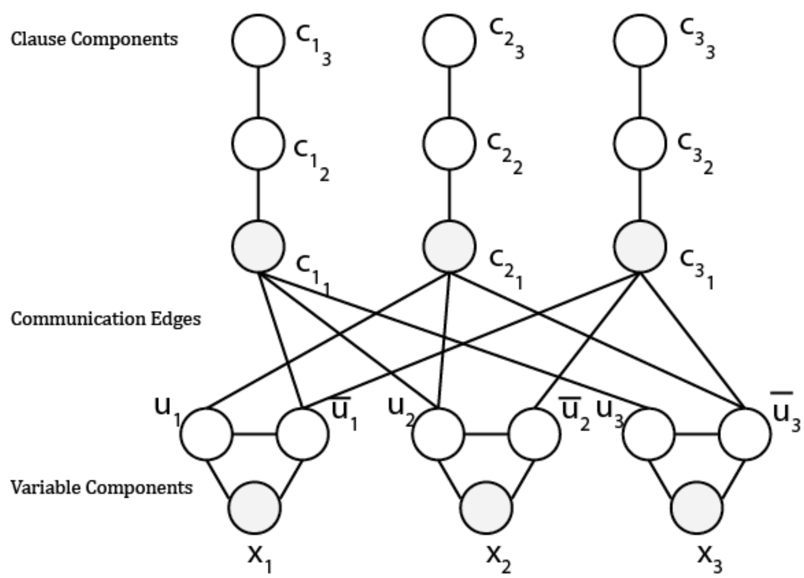

Figure 3. Example construction for Independent Restrained Domination

Figure 3 gives an example of these components in a construction for an instance of SAT where $U = \{u_1, u_2, u_3\}$ and $C = \{\{\overline{u_1}, u_2, u_3\}, \{u_1, u_2, \overline{u_3}\}, \{\overline{u_1}, \overline{u_2}, \overline{u_3}\}\}$. We show that there is a truth assignment for SAT if and only if G has an independent restrained dominating set.

Given a truth assignment $f: u \rightarrow \{T, F\}$ for the instance of SAT, we will find an independent restrained dominating set by selecting one vertex from each clause component, namely c_{j3}, and one vertex from each variable component, namely u_i if $f(u_i) =$ true or $\overline{u_i}$ if $f(\overline{u_i}) =$ false. Each vertex is dominated: the entire variable component is dominated by its selected element (u_i or $\overline{u_i}$), c_{j2} and c_{j3} by c_{j3}, and c_{j1} is dominated by a vertex corresponding to a true literal in the variable component. No two vertices in S are adjacent and every vertex in $V - S$ has a neighbor in $V - S$. In the example from Figure 3, let $f(u_1) =$ true, $f(u_2) =$ false and, $f(u_3) =$ true. A corresponding independent restrained dominating set is shown in Figure 4.

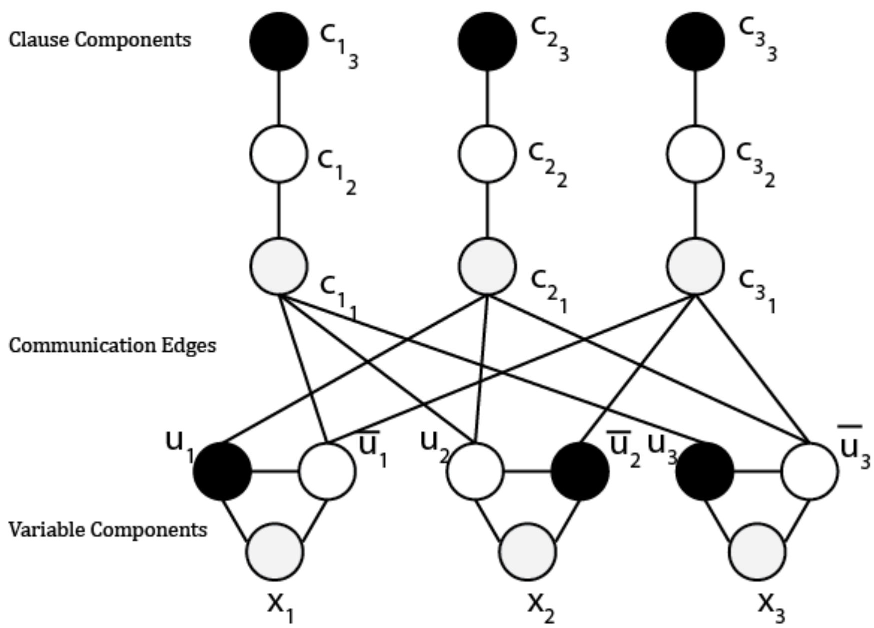

Figure 4. Sample independent restrained dominating set

Now we show that if G has a restrained dominating set, then there is a truth assignment for SAT. Suppose G has restrained dominating set S. Several observations can be made:

A) $c_{j3} \in S$, because no vertex with $deg(1)$ can be in $V - S$. See Lemma 3.1

B) $c_{j2} \notin S$, since S is independent.

C) $c_{j1} \notin S$, since c_{j2} must have a neighbor in $\{ V - S \}$.

D) Each c_{j1} vertex must be adjacent to a variable component vertex that is in S.

E) At most one vertex can be taken from each variable component because S is independent.

Given a independent restrained dominating set S, choose a truth assignment $f: u \rightarrow S$, by setting $f(u_i) =$ true if $u_i \in S$ and $f(u_i) =$ false if $u_i \notin S$. Every clause c_j will contain at least one true literal corresponding to the u_i or $\overline{u_i}$ that dominates c_{j1}.

Chapter 4: Minimal Restrained Dominating Set

In Chapter 3, we considered the Independent Restrained Dominating Set problem. For further study, we removed the requirement for independence and considered restrained dominating sets.

4.1 Definition & Characterization

Let G = (V, E) be a graph. A set $S \subseteq V$ is a dominating set if for every vertex $u \in V - S$, there exists $v \in S$ such that $uv \in E$. A set $S \subseteq V$ is a restrained dominating set if every vertex in $V - S$ is adjacent to a vertex in S and another vertex in $V - S$.

Lemma 4.1 *All vertices where deg(v) = 1 must be an element of S.*

Proof.

By the definition of restrained domination, any vertex in $V - S$ must also have a neighbor in $V - S$. A leaf vertex v where *deg(v)* = 1, that is v has only one neighbor, cannot have both a neighbor in S and a neighbor in $V - S$. Therefore, v must be an element of S.

Lemma 4.2 All graphs have a trivial restrained dominating set.

Proof.

Let G = (V, E) be a graph. A trivial restrained dominating set $S \subseteq V$ can be formed by setting $S = V$. Since all vertices are an element of S, all vertices are dominated, and since $V - S = \emptyset$, no vertex is required to have a neighbor in $V - S$.

Minimum restrained dominating sets have been studied [2] [5]. This section will consider a maximum version of the problem. Given that such a trivial solution always exists for restrained dominating sets, we choose to add a minimal condition to restrict out study to more interesting restrained dominating sets. A restrained dominating set S is a minimal restrained dominating set if no proper subset of S is also a restrained dominating set. Finding the largest minimal restrained dominating set was listed as an open problem by Hedetniemi [6].

4.2 Properties of Minimal Restrained Dominating Sets on Trees

Theorem 4.1. *A restrained dominating set S on a graph G=(V, E) is minimal if and only if every vertex v in S has at least one of the following properties:*

Property 1) $deg(v) = 1$

Property 2) v has a private neighbor with respect to S.

Property 3) $N(v) \subseteq S$ and $\forall v \in N(v)$: have property 1 or property 2.

Proof.

Let S be a restrained dominating set. If a vertex has degree 1, it must be an element of S because it cannot be adjacent to both a vertex in S and $V - S$. If a vertex v has a private neighbor with respect to S then it must be an element of S in order to dominate its private neighbor. If all of a vertex v's neighbors are in S and all of those neighbors have either property one or property two, then v must also be in S because it has no neighbor in $V - S$. If all vertices in S have at least one of the properties, then no vertex can be removed from S (otherwise S would no longer be a restrained dominating set).

Let S be a restrained dominating set for a graph $G=(V,E)$. Suppose $\exists v$:deg(v) > 1 and v does not have a private neighbor with respect to S and either $N(v) \not\subseteq S$ or $\exists w \in N(v)$: w does not have property 1 or w does not have property 2. We show S is not a minimal restrained dominating set.

Case 1: $deg(v) > 1$, v has no private neighbor with respect to S, $N(v) \not\subseteq S$. The set $S - \{v\}$ is still a restrained dominating set because no vertex depends on v to be dominated and v has at least one neighbor in S and one neighbor in $V - S$. Therefore, S is not minimal.

Case 2a: $N(v) \subseteq S$, $deg(v) > 1$, v has no private neighbor with respect to S, and $\exists w \in N(v)$, w does not have property 1, w does not have property 2, and w has only one neighbor in S, namely v. Note that since $deg(w) > 1$, w has a neighbor in $V - S$. Since w does not have property 2, no vertex in $N(w)$ depends on w to be dominated. Therefore, $S - \{w\}$ is a restrained dominating set, and S is not minimal.

Case 2b: $N(v) \subseteq S$, and $deg(v) > 1$, and $\exists w \in N(v)$: w does not have property 1, w does not have property 2, and w has more than one neighbor in S. Then, $S - \{v, w\}$ is a restrained dominating set, so S is not minimal. Note that since $deg(v) > 1$ and $\{N(v) - w\} \subseteq S$ then v is still dominated. Since v had no private neighbors with respect to S, all of its neighbors will still be dominated.

4.3 Dynamic Programming Algorithm

In order to apply the dynamic programming method, we must determine the base classes of graphs and their corresponding sets. These classes are illustrated in Figure 5 on page 24. Vertices that are a member of the subset are indicated with black shading. The classes are also described in Table 2 on page 25.

Next, we consider the effects of composing a graph from each class with another graph of each class. Table 3 gives the results of composing each of the base classes with each other for the twelve classes given in Figure 5. In this table, an ‘X’ is used to denote a composition which was not possible because the resulting tree-subset pair did not form a minimal restrained dominating set. Space limitations prevent us from providing a proof here for all possible compositions; however, we will give lemmas for several interesting compositions.

Lemma 4.3 *A class five subgraph cannot be composed as a child with parent subgraphs of classes one through six, class eight, and classes ten through twelve.*

Proof.

The root vertex of the class five subgraph must be given a private neighbor in the set. Since the root vertex in classes one through six, class eight, and classes ten through twelve is already dominated, the root vertex of the class five subgraph can be removed. Therefore the composition was not minimal.

Lemma 4.4 *Classes eleven and twelve cannot be composed, as a parent nor a child, with any class where the root is an element of S.*

Proof.

The root vertex of classes eleven and twelve cannot be dominated again. Therefore composing a subgraph of either class with another subgraph where the root is element of S creates a graph which is not minimal. Since the root is dominated by a vertex from a parent or child subgraph, then a child vertex of the root can be removed.

To ensure all classes were discovered, every combination of the distinguishing attributes between classes were considered. The distinguishing attributes were whether the root vertex is an element of S, if the root vertex has a neighbor in S, if the root vertex has a neighbor in $V - S$, and whether the root vertex was already dominated, was required as a private neighbor, or must be dominated by a later composition.

The algorithm follows the same general structure as the procedure given in Chapter 3, substituting the class compositions according to Table 3. A singleton vertex in S corresponds to class one, while a singleton vertex in $V - S$ corresponds to a class seven. Therefore, the initial vector is [1, -, -, -, -, -, 0, -, -, -, -, -]. The valid final classes for the root are 1, 2, 3, 5, 6, 10, and 12.

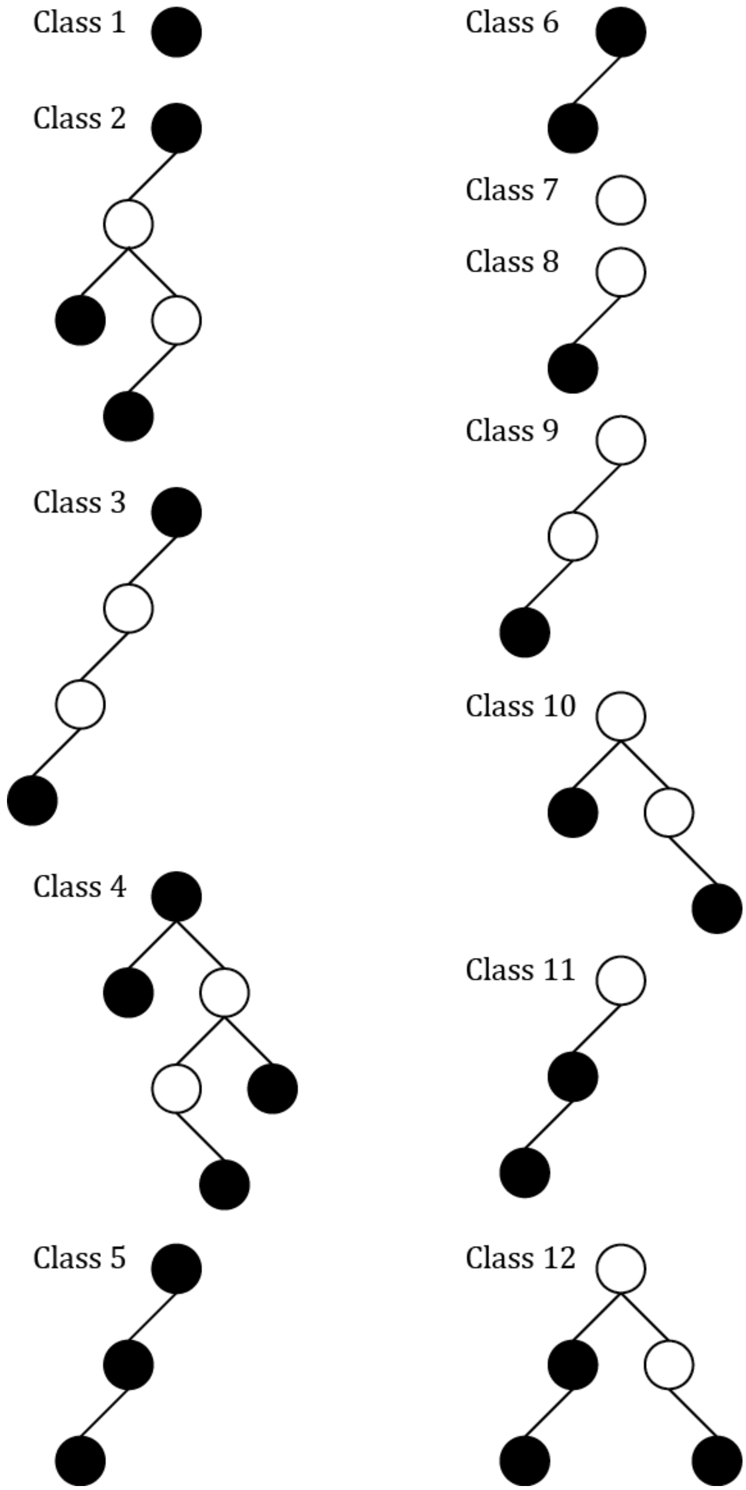

Figure 5. Base subgraphs for Minimal Restrained Dominating Set

Table 2. Descriptions of base subgraphs for Minimal Restrained Dominating Set

Class #	Description
1	$r \in S$, r has no neighbors
2	$r \in S$, r has a neighbor in $V - S$, r has no private neighbor except itself
3	$r \in S$, r has a neighbor in $V - S$, r has a private neighbor
4	$r \in S$, r has a neighbor in S and in $V - S$, r needs a private neighbor
5	$r \in S$, r has a neighbor in S, no neighbor in $V - S$, r must be used to dominate another vertex in $V - S$, r has no private neighbor, r's neighbor does not have property 1 nor property 2.
6	$r \in S$, r has a neighbor in S, no neighbor in $V - S$, all neighbors have properties 1 or 2
7	$r \notin S$, r has no neighbors, r is not dominated
8	$r \notin S$, r has a neighbor in S, no neighbor in $V - S$, r is not needed as a private neighbor
9	$r \notin S$, r has a neighbor in in $V - S$, r is not dominated
10	$r \notin S$, r has a neighbor in S and in $V - S$, r is dominated
11	$r \notin S$, r has a neighbor in S, but not $V - S$, r is a private neighbor, r cannot be dominated again
12	$r \notin S$, r has a neighbor in S and in $V - S$, r is a private neighbor, r cannot be dominated again

Table 3. Subgraph class compositions for Minimal Dominating Set

	1	2	3	4	5	6	7	8	9	10	11	12
1	6	X	6	X	X	5	X	X	3	2	X	X
2	4	X	4	X	X	4	X	X	3	2	X	X
3	3	X	3	X	X	3	X	X	3	3	X	X
4	4	X	4	X	X	4	X	X	5	4	X	X
5	4	X	4	X	X	4	X	X	3	4	X	X
6	6	X	6	X	X	6	X	X	5	4	X	X
7	8	8	8	11	11	11	X	9	X	9	9	9
8	8	8	8	X	X	X	X	10	X	10	10	10
9	10	10	10	12	12	12	X	9	X	9	9	9
10	10	10	10	X	X	X	X	10	X	10	10	10
11	X	X	X	X	X	X	X	12	X	12	12	12
12	X	X	X	X	X	X	X	12	X	12	12	12

4.4 NP-Completeness Proof For Minimal Restrained Dominating Set

Instance: Graph G, positive integer k.

Question: Does G have a minimal restrained dominating set of cardinality ≥ k?

Theorem 4.2. *Minimal Restrained Dominating Set is in the class NP.*

Proof.

A witness for Minimal Restrained Dominating Set is a set S of vertices on a graph G. We can verify that S is a minimal restrained dominating set in polynomial time by verifying that every vertex in S has one of the properties outlined in Theorem 4.1 and that every vertex in $V - S$ has a neighbor in S and a neighbor in $V - S$. Our transformation is from the well-known NP-complete Exact Cover by 3-Sets (X3C) problem [3].

Exact Cover by 3-Sets

Instance: A finite set $X = \{x_1, x_2, x_3, \dots, x_{3q}\}$, and a collection C of 3-element subsets of X.

Question: Does C contain an *exact cover* for X? In other words, does there exist a subset $C' \subseteq C$ such that every element of X occurs in exactly one member of C'?

Given an instance of X3C with a set X of variables, $|X| = 3q$, and a set C of 3-element subsets of X, $|C| = m$, generate a corresponding graph with components as follows:

For $\forall c_j \in C$ construct a subset component Cj such that:

$V(C_j) = \{ c_{j1}, c_{j2}, c_{j3}, c_{j4})$

$E(C_j) = \{ (c_{j1}, c_{j2}), (c_{j2}, c_{j3}), (c_{j3}, c_{j4})\}$

For $\forall x_i \in X$ construct an element component X_i such that:

$V(X_i) = \{ x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}, x_{i6})$

$E(X_i) = \{ (x_{i1}, x_{i2}), (x_{i2}, x_{i3}), (x_{i3}, x_{i4}), (x_{i3}, x_{i5}), (x_{i4}, x_{i6})\}$

Construct a set of communication edges CE such that:

$CE = \{(x_{i1}, c_{j1}) \mid x_i \in C_j\}$

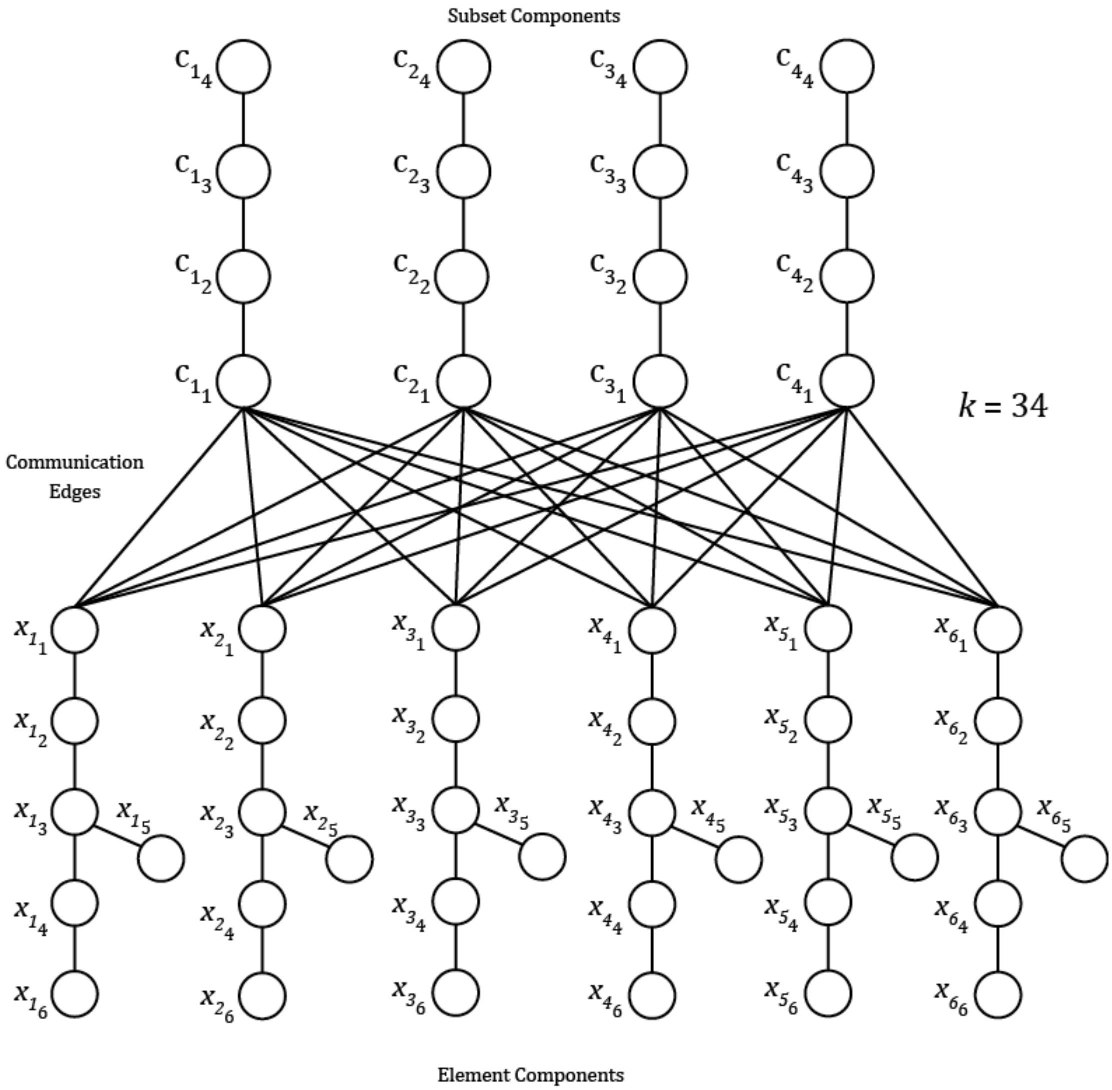

Figure 6. Example construction for Minimal Restrained Dominating Set

Set $k = 11q + 3m$. We show there is an exact cover for X if and only if G has a minimal restrained dominating set of cardinality $\geq 11q + 3m$. Figure 6 shows an example construction for a sample instance of SAT where $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ and $C = \{\{x_1, x_3, x_5\}, \{x_3, x_4, x_5\}, \{x_1, x_3, x_6\}, \{x_2, x_4, x_6\}\}$.

Suppose C has an exact 3-cover of C'. Then

$$\{\bigcup_{j=1}^{m} V(c_{j1}, c_{j4}) \mid C_j \in C'\} \cup \{\bigcup_{j=1}^{m} V(c_{j2}, c_{j3}, c_{j4}) \mid C_j \notin C'\} \cup \{\bigcup_{i=1}^{3q} V(x_{i3}, x_{i4}, x_{i5}, x_{i6})\}$$

is a minimal restrained dominating set of cardinality $11q + 3m$. In other words, to form the minimal restrained dominating set, take the vertices $\{c_{j2}, c_{j3}, c_{j4}\}$ for every subset element where the corresponding subset is not a member of the exact cover, and the vertices $\{c_{j1}, c_{j4}\}$ for every subset element where the corresponding subset is a member of the exact cover. Also include the vertices $\{x_{i3}, x_{i4}, x_{i5}, x_{i6}\}$ for every element subset. This construction can be accomplished in polynomial time. In our example, $C' = \{c_1, c_2\}$ is an exact cover, and the corresponding minimal restrained dominating set of size 34 is shown in Figure 7. Each vertex that is a member of the set is shaded black.

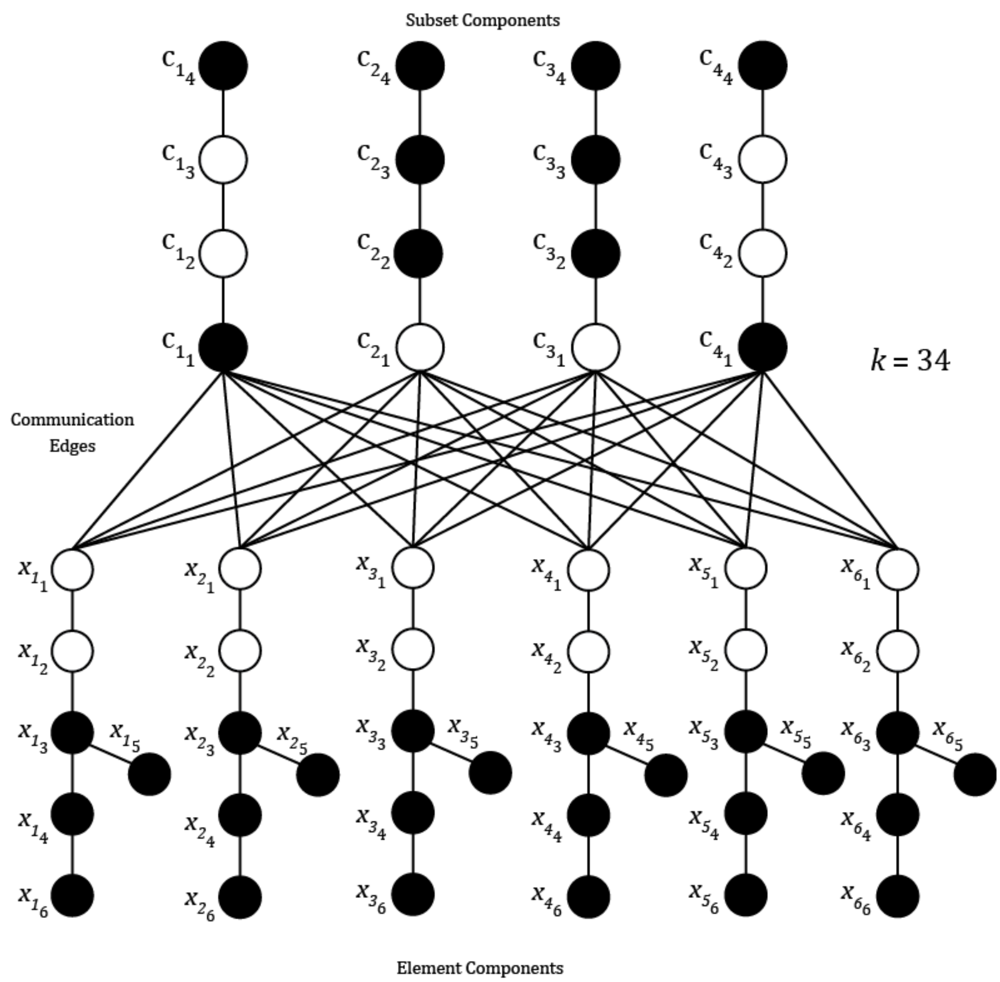

Figure 7. Sample minimal restrained dominating set.

Conversely, assume G has a minimal restrained dominating set S of cardinality $\geq k$. Several observations can be made regarding set S in G:

A) All leaf vertices are in S.

B) At most three vertices from each subset component are in S. If all four vertices in the component are in S then the middle two vertices, c_{j2} and c_{j3}, can be removed from the set (therefore the set was not minimal).

C) If three of the four vertices from each subset component are in S, then vertex c_{j1} must be the vertex not in S. If, instead, vertex c_{j1} is in S, then either vertex c_{j2} or c_{j3} must not be in S. In that case, however, neither c_{j2} nor c_{j3} would have a neighbor in $V - S$, and G would no longer be a restrained dominating set.

D) At most four vertices from each element component can be in S. All six vertices cannot be in S because vertices x_{i2} and x_{i3} could be removed; therefore the component was not minimal. Suppose five vertices from each element component are taken. If any vertex other than the root is left out, it will not have a neighbor in $V - S$. If the root is out, then vertices x_{i3} and x_{i4} could be removed as well.

E) If four vertices are taken from each element component, then those vertices must be {x_{i3}, x_{i4}, x_{i5}, x_{i6}}. Any other set of vertices from the element results in either a vertex not having a neighbor in $V - S$, or additional vertices that could be removed to make the set minimal. Figure 4 illustrates each of these cases. Each of the vertices surrounded by dotted lines is either a vertex in S that can be safely removed, or a vertex in $V - S$ which lacks a neighbor in $V - S$.

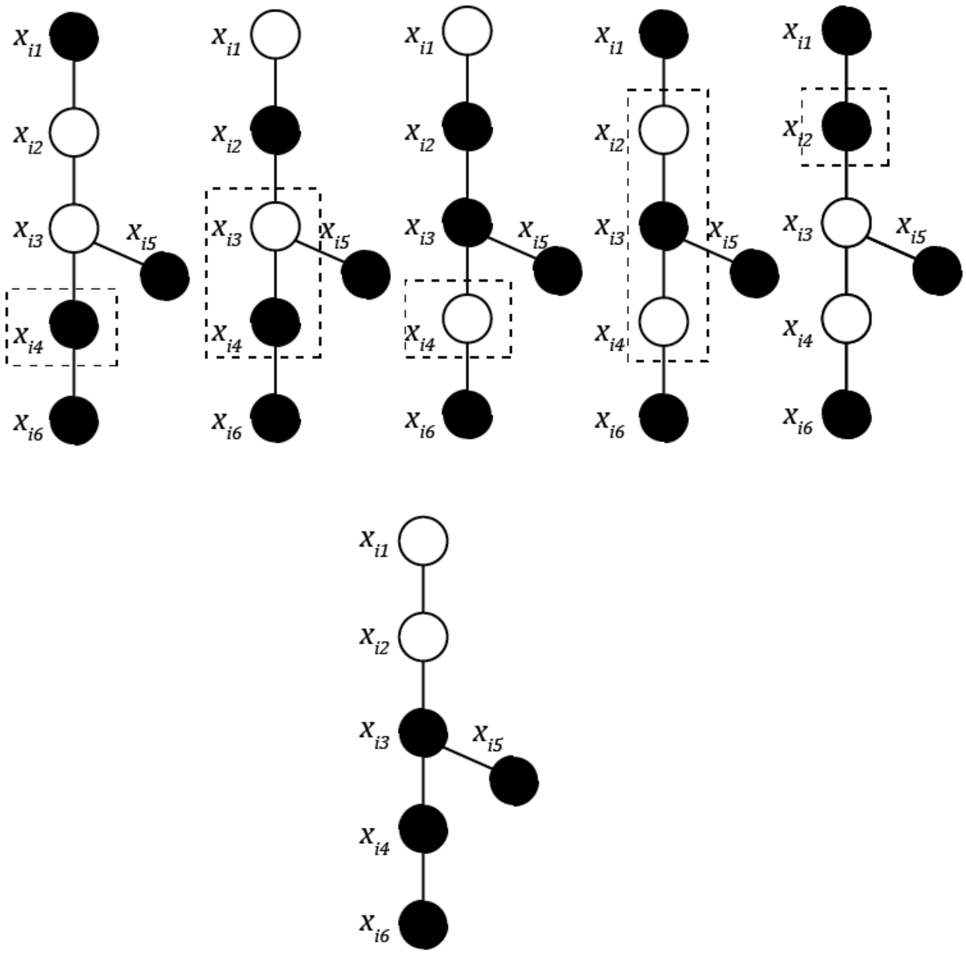

Figure 8. Element components where four vertices are taken

F) If four vertices are taken from each element component, then the root of the element, vertex x_{i1}, must be joined to a vertex in S from a subset component. That subset component cannot have three vertices in S.

Given these observations, let y be the number of element components where four vertices are taken. Each of these elements must have its root vertex, c_{j1}, dominated by the corresponding subset component. By the construction, each subset root is adjacent to only three element roots. Therefore, there must be at least $y/3$ subset components where just two vertices are taken. The number of components in of each type are given in Table 4.

Table 4. Number of occurrences for ech component type

Component Type	# of components
Element component where all four vertices are taken	y
Element components where less than four vertices are taken	$3q - y$
Subset components containing the root	$\geq y/3$
Subset components not containing the root	$\leq m - y/3$

Therefore, at most the number of vertices in S is given by:

$$2\left(\frac{y}{3}\right) + 3\left(m - \frac{y}{3}\right) + 4y\ + 3(y - 3q) =$$

$$9q + 3m + \frac{2y}{3}$$

To achieve $11q + 3m$, *y must be equal to 3q* and all element components have exactly four vertices in S, and exactly $\frac{y}{3} = q$ subset components have only two vertices in S. The subset components where the roots are taken correspond to subsets of C that form an exact cover for X.

Corollary 4.1 *Minimal restrained dominating set is NP-complete for bipartite graphs.*

Proof.

The graph in the construction is bipartite.

Chapter 5: Future Work

There are several areas to explore for future work. Our NP-completeness proof for minimal restrained domination serves to add further data points in the effort to determine what separates NP-complete problems from problems that can be solved with polynomial algorithms. Interestingly, for bipartite graphs, finding a largest minimal dominating set has a known polynomial algorithm [4]; however, as we have shown, adding the *restrained* requirement moves the problem into the set of NP-complete problems. On the other hand, our NP-completeness proof for the existence of independent restrained dominating set was not restricted to bipartite graphs. The upper minimal independent domination number is polynomial for bipartite graphs. Is independent restrained dominating set NP-complete for bipartite graphs?

The notion of an irredundant restrained set could be explored for both graphs and trees. In such a set S, every vertex in S has a private neighbor, and every vertex in $V - S$ has a neighbor that is also in $V - S$.

References

[1] Gary Chartrand, Linda Lesniak. Graphs & Digraphs (3rd Ed.). Wadsworth Publ. Co., 1996.

[2] G.S. Domke, J.H. Hattingh, S.T. Hedetniemi, R.C. Laskar and L.R. Markus. Restrained Domination in Graphs. In *Discrete Math*, 203:61-69, 1999.

[3] Michael R. Garey, David S. Johnson. Computers and Intractability: A guide to the Theory of NP-Completeness. Bell Laboratories, 1979.

[4] F. Harary. Graph Theory. Addison-Wesley, Reading, MA, 1969.

[5] Johannes H. Hattingh. Some Recent Results on Restrained Domination. The 23rd Mini-conference on Discrete Mathematics and Algorithms. Georgia State University, Atlanta, GA, USA, 2008.

[6] S.T. Hedetniemi. Unsolved algorithmic problems on trees. In *AKCE Internat. J. Graphs and Combinatorics*, 3:1-37, 2006.

[7] Teresa W. Haynes, Stephen Hedetniemi, Peter Slater. Fundamentals of Domination in Graphs. Marcel Dekker, New York, New York, 1998.

[8] S.T. Hedetniemi, R. Laskar. Bibliography on domination in graphs and some basic definitions of domination parameters. In Discrete Mathematics, 86(1-3):257-277,1990.

[9] J.A. Telle. Vertex partitioning problems: characterization, complexity and algorithms on partial k-trees. Ph.D. Thesis, University of Oregon, 1994.

[10] Thomas Victor Wimer. Linear Algorithms on K-Terminal Graphs. Ph.D. Dissertation. Clemson University, Clemson, SC, 1987.

Vita

Jeremy Booker was born in Raleigh, North Carolina, in September 1986. He graduated from East Wake High School in 2004 and entered college at North Carolina State University. He transferred to Appalachian State University in January 2005. In May 2008, he earned a Bachelor's of Science degree from Appalachian State University with *cum laude* honors in Computer Science with a minor in Business. Jeremy accepted a position with Appalachian's Office of Electronic Student Services in July 2008 and began pursing a Master's of Science degree in August 2009. He married Kristi in June 2010. He received this degree in December 2013. Jeremy and his wife currently live in West Jefferson, North Carolina.