Movable Dominating Sensor Sets in Networks

Jean Blair
Department of Electrical Engineering and Computer Science,
United States Military Academy,
West Point, NY, 10996, jean.blair@usma.edu

Ralucca Gera
Department of Applied Mathematics,
Naval Postgraduate School,
Monterey, CA, 93943, rgera@nps.edu

Steve Horton
Department of Mathematical Sciences,
United States Military Academy,
West Point, NY, 10996, steve.horton@usma.edu

Abstract

In this paper we consider 1-movable dominating sets, motivated by the use of sensors employed to detect certain events in networks, where the sensors have a limited ability to react under changing conditions in the network. A *1-movable dominating set* is a dominating set $S \subseteq V(G)$ such that for every $v \in S$, either $S - \{v\}$ is a dominating set, or there exists a vertex $u \in (V(G) - S) \cap N(v)$ such that $(S - \{v\}) \cup \{u\}$ is a dominating set. We present computational complexity results and bounds on the size of 1-movable dominating sets in arbitrary graphs. We also give a polynomial time algorithm to find minimum 1-movable dominating sets for trees. We conclude by extending this idea to k-movable dominating sets.

Keywords: Domination, Networks, Movable Sensors;
2000 Mathematical subject classification: 05C69.

1 Introduction

All graphs in this paper are simple. Let $G = (V(G), E(G))$ be a graph with $n = |V(G)|$ and $m = |E(G)|$. For any vertex $v \in V(G)$, the *open neighborhood* of v is the set $N(v) = \{u \mid uv \in E(G)\}$ and the *closed neighborhood* is the set $N[v] = N(v) \cup \{v\}$. Similarly, for any set $S \subseteq V(G)$, $N(S) = \cup_{v \in S} N(v)$ and $N[S] = N(S) \cup S$. A set S is a *dominating set* if $N[S] = V(G)$. The minimum cardinality of a dominating set of G is denoted by $\gamma(G)$. When S is a dominating set and $v \in S$, we say that there is a *sensor* at v, and that v is a *dominator*. Given a dominating set S, a vertex u is a *private neighbor* of $v \in S$ with respect to S if $N(u) \cap S = \{v\}$. If $u \in S$, we say it is an *internal* private neighbor of v, otherwise u is an *external* private neighbor of v. We use the notation (from [6]) $pn[v, S]$ to denote the set of private neighbors of $v \in S$ with respect to S. Similarly, we use $ipn[v, S]$ (and $epn[v, S]$) for the internal (and external) private neighbor set of v with respect to S.

The *degree* of a vertex v is the number of edges incident with v and is denoted by $\deg(v)$. The *minimum degree* of a graph $\delta(G) = \min_{v \in V(G)} \deg(v)$. The *distance*, $d(u, v)$, between two vertices u and v in G is the number of edges on a shortest path between u and v in G. The *eccentricity*, $e(v)$, of a vertex v is the largest distance from v to any vertex of G. The *radius* of G, $rad(G)$, is the smallest eccentricity in G. The *diameter* of G, $diam(G)$, is the largest eccentricity in G. Given disjoint graphs G and H, the *union* of G and H is the graph $G \cup H$ where $V(G \cup H) = V(G) \cup V(H)$ and $E(G \cup H) = E(G) \cup E(H)$. The *join* of G and H is the graph $G + H$ where $V(G + H) = V(G) \cup V(H)$ and $E(G + H) = E(G) \cup E(H) \cup \{uv : u \in V(G) \text{ and } v \in V(H)\}$.

This paper is organized in the following way. In the next section, we define a variation on dominating sets in which vertices in S are subject to being displaced. We call this new model a *1-movable* dominating set. Following, we establish the computational complexity of the problem on arbitrary graphs. In Section 4 we identify bounds on the 1-movable domination number for certain classes of graphs. We give a polynomial time algorithm to compute the 1-movable domination number of an arbitrary tree in Section 5. We conclude with some generalizations and directions for further research.

2 1-Movable Dominating Sets

Consider a network with sensors deployed at some vertices so that they can collectively detect certain events throughout the network. These sensors, for example, might be designed to detect electronic activity, human activity, or some natural phenomenon. If changing conditions at a vertex prevent the proper functioning of a sensor, it is natural to consider an alternate location for that sensor that is nearby and yet preserves the property that the sensors collectively cover the network. The conditions that prevent proper sensor operation might include adverse weather, electronic interference, or denial of service, etc. Motivated by this, we consider the following.

Definition 2.1 *A* 1-movable dominating set *is a dominating set $S \subseteq V(G)$ such that for every $v \in S$ there exists a vertex $u \in V(G) \cap N(v)$ such that $(S - \{v\}) \cup \{u\}$ is a dominating set.*

Note that Definition 2.1 does not distinguish between the case of $u \in S$ and $u \notin S$. In the former case, the set $S - \{v\}$ is itself a dominating set. The following definition is equivalent to Definition 2.1, and it explicitly distinguishes between these two cases. To simplify the explanations in this paper, the definition below will be used throughout.

Definition 2.2 *A* 1-movable dominating set *is a dominating set $S \subseteq V(G)$ such that for every $v \in S$, at least one of the following two conditions holds.*

1. *$S - \{v\}$ is a dominating set, or*
2. *there exists a vertex $u \in (V(G) - S) \cap N(v)$ such that $(S - \{v\}) \cup \{u\}$ is a dominating set.*

Informally, the idea is that every dominator is either not needed at all (condition 1), or can be replaced by a neighbor if an attack destroys its ability to provide domination so that the resulting set is also dominating for G (condition 2). Given a graph G, we will denote the cardinality of a smallest 1-movable dominating set by $\gamma_m^1(G)$. Since every 1-movable dominating set is a dominating set, $\gamma(G) \leq \gamma_m^1(G)$. Note that $\gamma_m^1(G)$ is undefined for any graph with an isolated vertex, and defined for all other graphs.

We start with a simple example. Consider P_3, the path on 3 vertices, with vertices v_1, v_2, and v_3 labeled consecutively. The set $S = \{v_2\}$ is a dominating set since $N[S] = V(G)$. However, S is not a 1-movable dominating set, nor is any other subset of $V(P_3)$ with cardinality 1. It is easy to verify that any choice of S satisfying $|S| \geq 2$ is a 1-movable dominating set for P_3. Thus, $\gamma(P_3) = 1$ but $\gamma_m^1(P_3) = 2$.

The following observations show that $\gamma(G)$ and $\gamma_m^1(G)$ can be equal, and that they can be arbitrarily far apart. Recall K_n is the complete graph on n vertices, and $K_{m,n}$ is the complete bipartite graph.

Observation 2.3 $\gamma(K_n) = \gamma_m^1(K_n) = 1$.

Observation 2.4 $\gamma(K_{1,n-1}) = 1$, *but* $\gamma_m^1(K_{1,n-1}) = n - 1$.

An immediate consequence of Observation 2.4 is that the value of $\frac{\gamma_m^1(G)}{\gamma(G)}$ can be arbitrarily large.

There are other classes of graphs for which $\gamma(G) = \gamma_m^1(G)$. Recall that the *corona* of an n-vertex graph G is the graph formed by adding n isolated vertices to G, and then connecting each new vertex to one from the original graph, so that each original vertex has one new private neighbor. The resulting graph $Cor(G)$ has $2n$ vertices and $|E(G)| + n$ edges. Since every dominating set of $Cor(G)$ is a 1-movable dominating set, we have the following lemma.

Lemma 2.5 *For any graph* G, $\gamma_m^1(Cor(G)) = \gamma(Cor(G))$.

It is also useful to observe that 1-movable dominating sets satisfy the property described below.

Observation 2.6 *If* S *is a 1-movable dominating set, then any* S^+ *satisfying* $S \subseteq S^+$ *is also a 1-movable dominating set.*

Note that without condition 1 in the Definition 2.2, Observation 2.6 fails to be true since for any graph with $\delta(G) \geq 1$, no vertex in the set $S = V(G)$ satisfies condition 2 (since $V(G) - S = \emptyset$).

2.1 Relationship to Secure Domination

In [3], the authors define a set S to be a *secure dominating set* if for every vertex $v \in V(G) - S$, there exists a vertex $u \in N(v) \cap S$ such that $(S - \{u\}) \cup \{v\}$ is a dominating set. They define the *secure domination number* $\gamma_s(G)$ to be the minimum size of a secure dominating set in G. There are some graphs for which all secure dominating sets are 1-movable dominating sets and vice-versa (for example, $K_{1,n}$). There are, however, other graphs that have 1-movable dominating sets that are not secure dominating sets. The following example demonstrates.

For the graph of Figure 1, $\{v_6, v_7\}$ is a 1-movable dominating set, since the sensor at v_6 can be moved to v_8, or the sensor at v_7 can be moved to v_9, and the resulting set ($\{v_7, v_8\}$ or $\{v_6, v_9\}$, respectively) dominates G. On the other hand, $\{v_6, v_7\}$ is not a secure dominating set since, for example, v_5 has no neighbor in $S = \{v_6, v_7\}$ that can transfer its sensor to v_5 and still form a dominating set for G.

Secure domination concepts were further studied in [2], and were extended in [5]. We revisit secure dominating sets and their relationship to 1-movable dominating sets in Section 4.

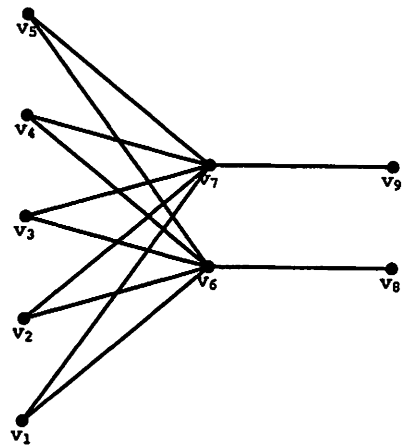

Figure 1: Example Graph

2.2 1-Movable Domination and Secure Domination as Games

Both 1-movable domination and secure domination can be thought of as two player games on graphs.

For 1-movable domination, player 1 selects a subset $S \subseteq V(G)$. Player 2 then chooses $v \in S$. If $S - v$ is a dominating set, player 1 wins (by condition 1 of Definition 2.2). Otherwise, player 1 then selects $u \in (V(G) - S) \cap N(v)$. If $(S - \{v\}) \cup \{u\}$ is a dominating set, player 1 wins. Otherwise player 2 wins.

For secure domination, player 1 selects $S \subset V(G)$. Player 2 then chooses $x \in V(G) - S$. Player 1 then selects $y \in x) \cap S$. If $(S - \{x\}) \cup \{y\}$ is a dominating set, player 1 wins. Otherwise player 2 wins.

3 Complexity

We can easily establish the complexity status of computing $\gamma_m^1(G)$. We start with two formal problem statements:

DOMINATING SET: Given a graph $G = (V(G), E(G))$ and a positive integer k, is there a subset $S \subseteq V(G)$ with $|S| \leq k$ such that S is a dominating set for G; that is, for all $u \in V(G) - S$ there is a $v \in S$ for which $\{u, v\} \in E(G)$? [4]

1-MOVABLE DOMINATING SET: Given a graph $G = (V(G), E(G))$ and a positive integer k, is there a subset $S \subseteq V(G)$ with $|S| \leq k$ such that S is a dominating set and for all $v \in S$, either $S - \{v\}$ is a dominating set or there is a $u \in V(G) - S$ for which $(S - \{v\}) \cup \{u\}$ is a dominating set?

Theorem 3.1 1-MOVABLE DOMINATING SET *is NP-complete.*

Proof. 1-MOVABLE DOMINATING SET is clearly in NP; given a subset S and an integer k it is easy to verify in polynomial time that S dominates G, $|S| \leq k$, and every $v \in S$ satisfies the specified property.

Our reduction is from DOMINATING SET. From an instance of DOMINATING SET we create an instance of 1-MOVABLE DOMINATING SET as follows. Given G and k, we use the join operation (see Section 1) to create $G^* = G + K_1$. We use the label x to denote the vertex in G^* corresponding to K_1. We claim that G has a dominating set S with $|S| \leq k$ if and only if G^* has a 1-movable dominating set S^* with $|S^*| \leq k$.

($\Rightarrow$) Suppose G has a dominating set S with $|S| \leq k$. Let S^* be the set S applied to the vertices of G^* that correspond to the vertices of S in G. Observe that $|S^*| = |S| \leq k$. Also observe that S^* dominates G^*: x is dominated by every member of S^*, and every other vertex in G^* is dominated because its corresponding vertex in G is dominated by S. Finally, to see that S^* is a 1-movable dominating set, observe that every dominator in S^* can move to x with the result being a dominating set for G^*.

($\Leftarrow$) Now suppose G^* has a 1-movable dominating set S^* with $|S^*| \leq k$. Either (1) $x \notin S^*$ or (2) $x \in S^*$. In case (1), we simply map S^* to S directly. Since S^* dominates G^*, S dominates G. In case (2), observe that since S^* is a 1-movable dominating set, x can be swapped out of S^* in exchange for some other vertex, say w, in $V(G^*) - S^*$ so that the set $(S^* - \{x\}) \cup \{w\}$ again dominates G^*. The vertices in G corresponding to $(S^* - \{x\}) \cup \{w\}$ form a dominating set for G of the required size. ∎

Despite the result of this section, there are cases where the value of $\gamma_m^1(G)$ can be bounded, or where it can be computed efficiently. The former is discussed in the next section, and the latter is considered in Section 5.

4 Bounds on the 1-Movable Domination Number

We first make an observation about disconnected graphs.

Observation 4.1 *Let G be a disconnected graph with no isolates, and let $G_1, G_2, \ldots, G_k$ be its components. Then*

$$\gamma_m^1(G) = \sum_{i=1}^{k} \gamma_m^1(G_i).$$

We next consider lower and upper bounds for the 1-movable domination number in connected graphs.

Theorem 4.2 *Let G be a connected graph of order $n \geq 2$. Then*

$$1 \leq \gamma_m^1(G) \leq n-1,$$

and the bounds are sharp.

Proof. The lower bound is immediate. For the upper bound, let $S = V(G) - \{x\}$ for an arbitrary vertex $x \in V(G)$. We show that S is a 1-movable dominating set. Since G is connected, x has a neighbor in S, so S dominates G. Now consider any $v \in S$. If $v \notin N(x)$, condition 1 of Definition 2.2 holds for v. On the other hand, if $v \in N(x)$, the set $(S-\{v\}) \cup \{x\}$ dominates G. Thus any set S with $|S| = n-1$ is a 1-movable dominating set. Sharpness follows from Observations 2.3 and 2.4. ∎

We next show that all the values between 1 and $n-1$ can be attained as a 1-movable domination number.

Theorem 4.3 *Let k and n be integers such that $1 \leq k < n$. Then there is a connected graph of order n and 1-movable domination number k.*

Proof. Consider the graph G obtained from K_{n-k+1} by adding $k-1$ isolated vertices $v_1, v_2, \ldots, v_{k-1}$. Then let $u \in V(K_{n-k+1})$ and add the edge $v_i u$ for each $1 \leq i \leq k-1$. Then $u, v_1, v_2, \ldots, v_{k-1}$ is a minimum 1-movable dominating set of order k. ∎

Figure 2 demonstrates for the case of $n = 8$ and $k = 4$.

We next present characterizations for the upper and lower bounds in Theorem 4.2.

Theorem 4.4 *Let G be a connected graph of order $n \geq 2$. Then*

$$\gamma_m^1(G) = 1 \iff G \text{ has two vertices of degree } n-1.$$

Proof. First, let G have two vertices of degree $n-1$, say x and y. Note $xy \in E(G)$. The set $S = \{x\}$ is a 1-movable dominating set of order 1 since both S and $(S - \{x\}) \cup \{y\}$ are dominating sets of G.

On the other hand, if $\gamma_m^1(G) = 1$, there is a vertex of degree $n-1$, say x, so that $S = \{x\}$ is a dominating set. Furthermore, there must be a vertex $y \neq x$ with $y \in N(x)$, such that $(S - \{x\}) \cup \{y\} = \{y\}$ is also a dominating set of G. Therefore y has degree $n-1$ and the result follows. ∎

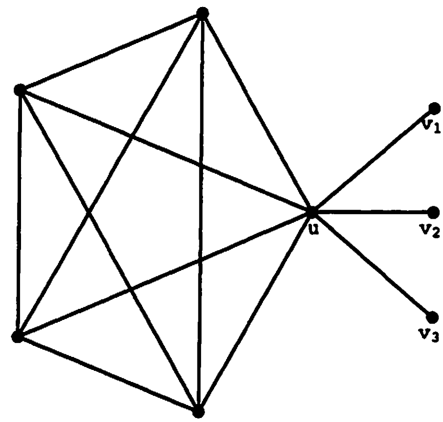

Figure 2: Demonstration of Theorem 4.3 for $n = 8$ and $k = 4$

Theorem 4.5 *Let G be a connected graph of order $n \geq 2$. Then*

$$\gamma_m^1(G) = n - 1 \iff G = K_{1,n-1}.$$

Proof. The "only if" direction follows immediately from Observation 2.4. For the "if" direction, we will show that if $G \neq K_{1,n-1}$, then $\gamma_m^1(G) < n-1$. For $n = 2$, no graph satisfies the premise, so the statement is true. For $n = 3$, only K_3 needs consideration, and $\gamma_m^1(K_3) = n - 2 = 1$. So we consider $n \geq 4$. Let $x_1, x_2, \ldots, x_t$ be a longest vertex disjoint path in G. Since $G \neq K_{1,n-1}$, we know $t \geq 4$. Now the set $S = V(G) - \{x_1, x_t\}$ is a 1-movable dominating set of order $n - 2$. ∎

For bipartite graphs, we have the following results.

Corollary 4.6 *Let G be a connected bipartite graph of order $n \geq 3$. Then*

$$2 \leq \gamma_m^1(G) \leq n - 1,$$

and the bounds are sharp.

Proof. The first inequality follows from Theorem 4.4, along with the observation that any graph with $n \geq 3$ having two vertices of degree $n - 1$ must contain a triangle, and therefore cannot be bipartite. Sharpness of the first inequality follows from the fact that $\gamma_m^1(K_{2,n-2}) = 2$. The second inequality along with the fact that it is sharp follows immediately from Theorem 4.2. ∎

We also show that the full range of values established in Corollary 4.6 can be attained as the 1-movable domination number of a bipartite graph.

Theorem 4.7 *Let k and n be integers such that $2 \le k < n$. There is a connected bipartite graph of order n and 1-movable domination number k.*

Proof. If $k = n-1$, then by Observation 2.4 the graph $G = K_{1,n-1}$ fulfills the requirement. So assume $k \le n-2$ and for any specific pair k and n build a graph G from the $K_{2,n-2}$ with vertices $\{x, y, v_1, v_2, \ldots, v_{n-2}\}$ by removing the edge xv_i for each i, $1 \le i \le k-1$. Since y remains adjacent to every v_i and x is adjacent to at least v_{n-2}, G is connected. We must now show that $\gamma_m^1(G) = k$.

First, we show that the set $S = \{x, y, v_1, \ldots, v_{k-2}\}$ is a 1-movable dominating set of cardinality k. Each $v_i \in S$ satisfies condition 1 of Definition 2.2, since y is adjacent to every v_i. The sensor at x can be moved to any v_j with $j \ge k$. Finally, the sensor at y can be moved to v_{k-1}. Thus, condition 2 of Definition 2.2 holds for both x and y. Thus, S is a 1-movable dominating set of cardinality k.

Finally, we must show that $\gamma_m^1(G) \ge k$. Following Observation 2.4, any 1-movable dominating set of G must include $k-1$ vertices in $\{y, v_1, \ldots, v_{k-1}\}$. Moreover, since x is not adjacent to any vertex in $\{y, v_1, \ldots, v_{k-1}\}$, the 1-movable dominating set must include at least one more than the required $k-1$ vertices. It follows that $\gamma_m^1(G) \ge k$. ∎

We can also bound $\gamma_m^1(G)$ in terms of other domination parameters.

Theorem 4.8 *Let G be a connected graph of order $n \ge 2$. Then $\gamma(G) \le \gamma_m^1(G) \le \gamma_s(G)$.*

Proof. Since all 1-movable dominating sets are by definition dominating sets, the lower bound is immediate. For the upper bound, we will show that all secure dominating sets are 1-movable dominating sets. Suppose S is a secure dominating set. Consider $v \in S$. If $N(v) \cap (V(G) - S) = \emptyset$, then, since G is connected, v has a neighbor in $S - \{v\}$. Therefore, $S - \{v\}$ is a dominating set and so condition 1 of Definition 2.2 holds. On the other hand, let $Q = N(v) \cap (V(G) - S) \neq \emptyset$. Since S is a secure dominating set, every $q_i \in Q$ has a neighbor in S, say s_i, such that $(S - \{s_i\}) \cup \{q_i\}$ is a dominating set. If $v = s_i$ for some i, then $(S - \{v\}) \cup \{q_i\}$ is a dominating set and condition 2 of Definition 2.2 holds for v. Otherwise, every q_i is adjacent to some other vertex in $S - \{v\}$ and the set $(S - \{v\}) \cup \{q_i\}$ for any $q_i \in Q$ is a dominating set and condition 2 of Definition 2.2 holds for v. Therefore S is a 1-movable dominating set. ∎

In Section 2 we saw that there are graphs for which these inequalities can be strict, and graphs for which the bounds are tight. We can extend the upper bound in Theorem 4.8 by finding an upper bound for $\gamma_s(G)$ in terms of yet another domination parameter. Recall that S is a 2-dominating set of G if $\forall v \in V(G)-S$, $|N(v) \cap S| \ge 2$. Informally, this means that every vertex that is not a dominator must be adjacent to two vertices that are dominators. Given a graph G, we denote the order of a smallest 2-dominating set by $\gamma_2(G)$.

Theorem 4.9 *Let G be a connected graph of order $n \geq 2$. Then $\gamma_s(G) \leq \gamma_2(G)$.*

Proof. We establish the result by showing that every 2-dominating set is a secure dominating set. Consider a graph G and a 2-dominating set S. Consider $v \in V(G) - S$. Since S is 2-dominating, there must be two vertices $x, y \in N(v) \cap S$. The set $(S - \{x\}) \cup \{v\}$ then dominates G because x is dominated by v, and every vertex in $N(v)$ is either a dominator, or it is dominated by a vertex in $S - \{v\}$. Therefore S is a secure dominating set and the result follows. ∎

The next bound is based on the clique partition number $\bar{\chi}(G)$. Recall a *clique partition* of a graph G is a partition of $V(G)$ into sets $V_1, V_2, ..., V_k$ so that the subgraph induced by each V_i is isomorphic to the clique on $|V_i|$ vertices. The clique partition number of a graph G is the minimum number of cliques that G can be partitioned into.

Theorem 4.10 *Let G be a connected graph of order $n \geq 2$. Then $\gamma_m^1(G) \leq \bar{\chi}(G)$.*

Proof. Consider a clique partition $V_1, V_2, ..., V_k$ of G. Select an arbitrary vertex x_i from each V_i. We show that the resulting set $S = \{x_1, x_2, ..., x_k\}$ is a 1-movable dominating set. For each V_i, either $|V_i| = 1$ or $|V_i| > 1$. In the latter case, we note that the sensor at x_i can be moved to another vertex in V_i. In the former case, the sensor at x_i can be moved to any adjacent vertex in G or, if $N(x_i) \cap (V(G) - S) = \emptyset$, the sensor at x_i is redundant and can be removed. ∎

The last result of this section determines the 1-movable domination number for paths.

Theorem 4.11 *Let P_n be a path on n vertices. Then $\gamma_m^1(P_n) = \lceil \frac{2n}{5} \rceil$.*

Proof. Label the vertices of P_n in the usual way. First we note that $\gamma_m^1(P_n) \leq \lceil \frac{2n}{5} \rceil$ since the set $S = \{v_i | i \equiv 2 (\text{mod } 5) \text{ or } 4 (\text{mod } 5)\}$ is a 1-movable dominating set. Now suppose $\gamma_m^1(P_n) < \lceil \frac{2n}{5} \rceil$. By the pigeonhole principle, there exists an induced subgraph isomorphic to P_5 that has at most one dominator. Label the vertices of this subgraph with $\{v_1, v_2, v_3, v_4, v_5\}$. Note that $v_3 \in S$ since other choices fail to dominate either v_2 or v_4. Now Definition 2.2 does not hold at v_3. ∎

5 1-Movable Domination on Trees

In this section, we establish a 1-movable programming algorithm to efficiently compute $\gamma_m^1(T)$ for an arbitrary tree T. Recall a tree is a connected graph with no cycles. We start by observing that Definition 2.2 has an alternative characterization in the case of trees. This turns out to be useful in computing $\gamma_m^1(T)$.

Theorem 5.1 *Given a tree T with $n \geq 2$ and a dominating set S for T, S is a 1-movable dominating set if and only if $|epn[v, S]| \leq 1$ for all $v \in S$.*

Proof. ($\Leftarrow$) Consider $v \in S$. The only vertices in T that are not dominated by $S - \{v\}$ are those in $epn[v, S]$, and possibly v itself. Since $|epn[v, S]| \leq 1$, either $|epn[v, S]| = 1$ or $|epn[v, S]| = 0$. In the former case, we can move the sensor at v to the vertex in $epn[v, S]$; it dominates both v and $epn[v, S]$. Therefore $(S - \{v\}) \cup epn[v, S]$ dominates T and condition 2 of Definition 2.2 holds for v. In the latter case, we can move the sensor at v to any vertex in $N(v)$ and again the resulting set dominates T. Therefore, S is a 1-movable dominating set for T.

($\Rightarrow$) Now suppose $|epn[v, S]| > 1$ for some $v \in S$; let x and y be vertices in $epn[v, S]$. The sensor at v cannot be deleted since $N[S - \{v\}] \cap epn[v, S] = \emptyset$. Then without loss of generality we must move the sensor at v to x. Since xv and yv are edges in T, xy cannot be an edge or T contains a cycle. Thus, the sensor at x fails to dominate y, so S cannot be a 1-movable dominating set. ■

Note that Theorem 5.1 cannot be extended to arbitrary graphs since, for example, the graph K_3 has a 1-movable dominating set consisting of a single vertex, but that vertex has two external private neighbors. That is, $|epn[v, S]| \leq 1$ is a sufficient condition for S to be a 1-movable dominating set in any graph (since the ($\Leftarrow$) portion of the proof does not rely on any special structure in T), but the converse fails.

Theorem 5.1 provides the insight needed to create a 1-movable programming algorithm for computing $\gamma_m^1(T)$ when T is a tree.

5.1 Algorithm for Trees

The algorithm given in Figure 3 shows the generic bottom-up algorithm for processing a tree. Our algorithm will be an instantiation of BOTTOM-UP with appropriate procedures for PROCESS and BEST-AT-ROOT.

The general approach is to compute $f(v)$ for each vertex v in a rooted tree T by considering the subtree T_v of T induced by v and its descendents.

```
Algorithm BOTTOM-UP
Input: A tree T = (V(T), E(T)) with n ≥ 2.
Output: Generic optimal value for T.
begin
    Root T at an arbitrary vertex r;
    Leaves ← {v ∈ V | (deg(v) = 1) and (v ≠ r)};
    Child-Count(r) ← deg(r);
    for each v ∈ V(T) − {r} do
        Child-Count(v) ← deg(v) − 1;
    Ready ← Leaves;

    while Ready ≠ ∅ do
        Remove a vertex v from Ready;
        PROCESS(v, Children(v), f(x) for each x ∈ Children(v));
        Child-Count(Parent(v)) ← Child-Count(Parent(v)) − 1;
        if Child-Count(Parent(v)) = 0 then
            Ready ← Ready + Parent(v);
    end while
    return BEST-AT-ROOT(T, r, f(r));
end
```

Figure 3: Generic bottom-up algorithm for a tree.

Each $f(v)$ value is a five-tuple (see Table 1). Each component of the five-tuple, $f(v)[z]$, is the number of vertices in a smallest 1-movable dominating set S_v of T_v with the property z, or in the cases of OUTUD and OUTUU, a smallest set that satisfies Definition 2.2 in T_v everywhere except at v.

Table 1: 1-movable domination properties for tree T_v

Component	Properties
$f(v)[\text{IN0}]$:	$v \in S_v$ and $\lvert epn[v, S_v]\rvert = 0$
$f(v)[\text{IN1}]$:	$v \in S_v$ and $\lvert epn[v, S_v]\rvert = 1$
$f(v)[\text{OUTC}]$:	$v \notin S_v$ and $\exists w \in N(v) \cap S_v$: $epn[w, S_v] = \{v\}$ or $\lvert N(v) \cap S_v\rvert \geq 2$
$f(v)[\text{OUTUD}]$:	$v \notin S_v$ and $\nexists w \in N(v) \cap S_v$: $epn[w, S_v] = \{v\}$ and $\lvert N(v) \cap S_v\rvert = 1$
$f(v)[\text{OUTUU}]$:	$v \notin S_v$ and $N(v) \cap S_v = \emptyset$

Note that properties IN0, IN1, and OUTC partition the possible 1-movable dominating sets of a tree T_v rooted at v. If S_v is a 1-movable dominating set for T_v, either $v \in S_v$ or $v \notin S_v$. In the former case, by Theorem 5.1, either v has no external private neighbors with respect to S_v (IN0) or it has one (IN1). In the case that $v \notin S_v$, either v must have a neighbor $w \in S_v$

so that $epn[w, S_v] = \{v\}$, or v must have *two* neighbors in S_v (these are both included in OUTC). The properties OUTUD and OUTUU characterize the cases of sets $S_v \subseteq V(T_v)$ that are *not* 1-movable dominating sets for T_v but are such that if v is dominated by its parent $p(v)$, then $S_v \cup \{p(v)\}$ is a 1-movable dominating set for $T_{p(v)}$. The characteristic OUTUU represents cases where, in T_v, $|S_v \cap N(v)| = 0$ so that $epn[p(v), S_{p(v)}] = \{v\}$ must hold for $S_{p(v)} \supset S_v$. Similarly, the characteristic OUTUD represents cases where, in T_v, $|S_v \cap N(v)| = 1$ so that $v \notin epn[p(v), S_{p(v)}]$ for any $S_{p(v)} \supset S_v$.

Now we show how to compute $f(v)$, given a tree T_v rooted at v. Since the properties IN0, IN1 and OUTC represent 1-movable dominating sets for T and OUTUD and OUTUU do not, when we compute $f(r)$ at the root r of our final tree T, the only portions of $f(r)$ that are candidates for $\gamma_m^1(T)$ are $f(r)[\text{IN0}], f(r)[\text{IN1}]$, and $f(r)[\text{OUTC}]$. The Algorithm BEST-AT-ROOT follows from this idea.

5.2 Details of the Algorithm

In this section, we describe technical details of the algorithms described in the previous section. These algorithms generally follow approaches described in Wimer, Hedetniemi, and Laskar [7] and Borie, Parker, and Tovey [1]. Informally, the Algorithm BOTTOM-UP with PROCESS and BEST-AT-ROOT operates in the following way. Again we let T_v denote the subgraph of the (rooted) instance tree T induced by v and its descendants, and let $C(v)$ denote the set of children of v. At each leaf node, $f(\text{leaf})$ is defined according to $f(\text{leaf}) \leftarrow < 1, \infty, \infty, \infty, 0 >$. For example, $f(\text{leaf})[\text{IN0}] = 1$ because the cost (the number of dominators used) of putting a leaf node in S is 1. It is clear that $f(\text{leaf})[\text{IN1}] = f(\text{leaf})[\text{OUTC}] = f(\text{leaf})[\text{OUTUD}] = \infty$ because leaf nodes have no descendants. Finally, $f(\text{leaf})[\text{OUTUU}] = 0$ because no dominators are used in this case. Then at each node v that is not a leaf, $f(v)$ is computed once $f(x)$ is known for each child x of v. When the entire tree T is processed in this way, we apply BEST-AT-ROOT to determine $\gamma_m^1(T_r) = \gamma_m^1(T)$.

The first component of $f(v)$, $f(v)[\text{IN0}]$, represents the number of dominators used in a 1-movable dominating set S_v of T_v where $v \in S_v$ and v has no external private neighbors in T_v with respect to S_v. The notation we use IN0 is short for "in S, with 0 external private neighbors". In the computation, the 1 accounts for the dominator at v, and then we add to that the sum over all of the children of v, allowing all cases where $|epn[v, S_v]| = 0$ holds. This excludes $f(x)[\text{OUTUU}]$ for $x \in C(v)$ since a vertex x that is not dominated from below requires domination from a private dominator for the result to lead to a 1-movable dominating set in T. Therefore, for each child x of v, we take the smallest $f(x)$ value from the other components.

Algorithm PROCESS
Input: A vertex v, its set of children $C(v)$, and $f(x)$ for each $x \in C(v)$.
Output: $f(v)$.
begin
  if $v \in Leaves$, **then** $f(v) \leftarrow \langle 1, \infty, \infty, \infty, 0 \rangle$;
  else

$$f(v)[\text{IN0}] \leftarrow 1 + \sum_{x \in C(v)} \min \{ f(x)[\text{IN0}], f(x)[\text{IN1}], f(x)[\text{OUTC}], f(x)[\text{OUTUD}] \};$$

$$f(v)[\text{IN1}] \leftarrow 1 + \min_{x \in C(v)} \left\{ f(x)[\text{OUTUU}] + \sum_{\substack{y \in C(v) \\ y \neq x}} \min \left\{ \begin{array}{l} f(y)[\text{IN0}] \\ f(y)[\text{IN1}] \\ f(y)[\text{OUTC}] \\ f(y)[\text{OUTUD}] \end{array} \right\} \right\};$$

$$f(v)[\text{OUTC}] \leftarrow$$

$$\min \left\{ \begin{array}{l} \min\limits_{x \in C(v)} \left\{ f(x)[\text{IN0}] + \sum\limits_{\substack{y \in C(v) \\ y \neq x}} \min \left\{ \begin{array}{l} f(y)[\text{IN0}] \\ f(y)[\text{IN1}] \\ f(y)[\text{OUTC}] \end{array} \right\} \right\} \\ \min\limits_{\substack{x \in C(v) \\ y \in C(v) \\ x \neq y}} \left\{ f(x)[\text{IN1}] + f(y)[\text{IN1}] + \sum\limits_{\substack{z \in C(v) \\ z \neq x \\ z \neq y}} \min \left\{ \begin{array}{l} f(y)[\text{IN0}] \\ f(y)[\text{IN1}] \\ f(y)[\text{OUTC}] \end{array} \right\} \right\} \end{array} \right\};$$

$$f(v)[\text{OUTUD}] \leftarrow \min_{x \in C(v)} \left\{ f(x)[\text{IN1}] + \sum_{\substack{y \in C(v) \\ y \neq x}} f(y)[\text{OUTC}] \right\};$$

$$f(v)[\text{OUTUU}] \leftarrow \sum_{x \in C(v)} f(x)[\text{OUTC}];$$

end

Figure 4: PROCESS

Algorithm BEST-AT-ROOT
Input: A tree T, a root vertex r, and $f(r)$ computed by PROCESS.
Output: $\gamma_m^1(T)$.
begin
  $\gamma_m^1(T) = \min\{f(r)[\text{IN0}], f(r)[\text{IN1}], f(r)[\text{OUTC}]\}$
end

Figure 5: BEST-AT-ROOT

The second component $f(v)[\text{IN1}]$ represents the number of dominators used in a 1-movable dominating set S_v of T_v where $v \in S_v$ and v has one external private neighbor in T_v with respect to S_v. The notation we use IN1 is short for "in S, with 1 external private neighbor". In the computation, the 1 accounts for the dominator at v. In the other portion of the computation, since $|epn[v, S_v]| = 1$, one child of v must be OUTUU to demand private domination from v. For the other children of v, any component of the five-tuple can be used *except* OUTUU.

The other three components of the five-tuple $f(v)$ represent cases where $v \notin S_v$. We partition these cases into classes depending on the answers to two questions. First, we compute $|N(v) \cap S_v|$ and note that it either equals 0, equals 1, or is at least 2. The second question can be answered with a "yes" or a "no": does there exist a neighboring sensor whose only external private neighbor is v? If $|N(v) \cap S_v| \geq 2$, we do not need to consider the answer to the second question. Regardless of private neighbor relationships, if v has two children that are in S_v, the domination provided to v by one of these children will be present even if the sensor at the other child is moved. Thus v is "out" of S_v, but "covered" from its children in terms of 1-movable domination. This gives rise to the notation OUTC. The other case that leads to OUTC is where $|N(v) \cap S_v| = 1$ and $\exists w \in N(v) \cap S_v : epn[w, S_v] = \{v\}$. Here, even though there is only one sensor in the children of v, that sensor's only external private neighbor is v which again means v is "covered" from below. This again leads to OUTC. The component $f(v)[\text{OUTC}]$ therefore represents the number of dominators used in a 1-movable dominating set S_v of T_v where $v \notin S_v$. We compute $f(v)[\text{OUTC}]$ by considering the two situations outlined above separately and then using the minimum.

If $|N(v) \cap S_v| = 1$ and $\nexists w \in N(v) \cap S_v : epn[w, S_v] = \{v\}$, the vertex v has exactly one neighboring sensor. Let z be the vertex containing that sensor. Since $\nexists w \in N(v) \cap S_v : epn[w, S_v] = \{v\}$, we know $|epn[z, S_v]| \geq 2$. In this case, we say that v is "out" of S_v and "uncovered" in that if we are forced to move the sensor at z, some vertex in $epn[z, S_v]$ will be undominated. We also say that v is "dominated" because z provides domination. This gives rise to the notation OUTUD (OUT, Uncovered, Dominated). The component $f(v)$[OUTUD] represents the number of dominators used in a set S_v that is a 1-movable dominating set for $T_v - \{v\}$, and in which $|N(v) \cap S_v| = 1$. We compute $f(v)$[OUTUD] by forcing one of v's children to be IN1 (hence providing domination to v but not coverage in terms of 1-movable domination) and the other children to be OUTC.

If $|N(v) \cap S_v| = 0$, there are no neighboring sensors, and we are in class OUTUU (OUT, Uncovered, Undominated). The component $f(v)$[OUTUU] therefore represents the number of dominators used in a set S_v that is a 1-movable dominating set for $T_v - \{v\}$, and in which $|N(v) \cap S_v| = 0$. The computation of $f(v)$[OUTUU] is simple; each child of v must be not in S_v and must also be "covered" from below. Therefore we sum the $f(x)$[OUTC] components for $x \in C(v)$.

A complete example of this computation is given in Section 5.4.

5.3 Complexity

Let $n = |V(T)|$ and note that since T is a tree $|E(T)| = n-1$. In this section we describe how to implement Algorithm BOTTOM-UP with PROCESS and BEST-AT-ROOT in $O(n)$ time.

Clearly BEST-AT-ROOT is a constant time operation and the initialization before the while loop in BOTTOM-UP requires only $O(n)$ time. We will show that each call to PROCESS takes time proportional to the number of children of its parameter v. Since all other operations in the while loop are constant time operations, it will then follow that the total time for the while loop is $O(|E(T)|) = O(n)$, thereby proving the desired result.

We begin by noting that in PROCESS, computation of $f(v)$[IN0] and $f(v)$[OUTUU] are straight-forward $O(|C(v)|)$ operations. The efficiency of the remaining computations, i.e., $f(v)$[IN1], $f(v)$[OUTC], and $f(v)$[OUTUD], rely on the simple mathematical fact that we can add a value and its inverse to a total without changing the value of the total. For example, in the computation of $f(v)$[OUTUD], for any one child $x \in C(v)$

$$f(x)[\text{IN1}] + \sum_{\substack{y \in C(v) \\ y \neq x}} f(y)[\text{OUTC}] = f(x)[\text{IN1}] - f(x)[\text{OUTC}] + \sum_{y \in C(v)} f(y)[\text{OUTC}].$$

Thus, if we pre-compute $\sum_{y \in C(v)} f(y)$[OUTC], we can compute the value of $f(v)$[OUTUD] by finding the minimum, across all $x \in C(v)$, of $f(x)$[IN1] $-$ $f(x)$[OUTC] and then adding that minimum to the pre-computed sum. This makes computation of $f(x)$[OUTUD] two sequential $O(|C(v)|)$ time operations, for a total of $O(|C(v)|)$ time. Similarly, computing $f(v)$[IN1] requires only $O(|C(V)|$ time.

Consider now the computation of $f(v)[\text{OUTC}]$ where we must choose the minimum of two minimum values. We can compute the first minimum value as described above in $O(|C(v)|)$ time. For the second minimum value we must be able to pick a *pair* $x, y \in C(v)$ that minimizes a sum, rather than a single $x \in C(v)$ as above. But this is easy since we need only find the smallest and second smallest $f(x)[\text{IN1}] - \min\{f(x)[\text{IN0}], f(x)[\text{IN1}], f(x)[\text{OUTC}]\}$ values among the children $x \in C(v)$ and then compute the sum as above. It follows that $f(v)[\text{OUTC}]$ can also be computed in $O(|C(v)|)$ time.

We have shown that by using $O(|C(v)|)$ time to pre-compute select sums across the children of v, Algorithm PROCESS can compute all five $f(v)$ vector values in $O(|C(v)|)$ time. We therefore have the following result.

Theorem 5.2 *Algorithm* BOTTOM-UP *with* PROCESS *and* BEST-AT-ROOT *computes* $\gamma_m^1(T)$ *for any tree* T *in* $O(n)$ *time.*

5.4 An Example

Consider the rooted tree shown in Figure 6 below. For the leaf nodes that are not the root we can begin by noting that $f(v_1) = f(v_2) = f(v_3) = f(v_4) = f(v_5) = f(v_6) = \langle 1, \infty, \infty, \infty, 0 \rangle$. The remaining calculations are in Table 2.

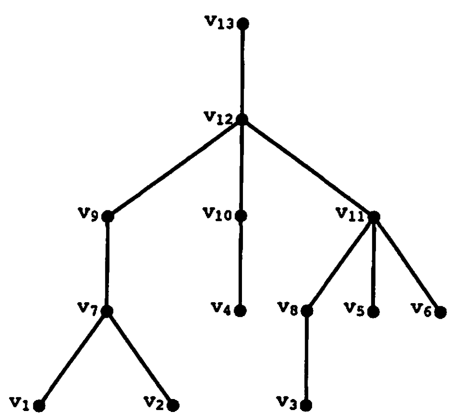

Figure 6: Example Tree

As an example, consider the calculation of $f(v_{11})$. The children of v_{11} in T are v_5, v_6, and v_8. Thus $C(v_{11}) = \{v_5, v_6, v_8\}$. Therefore, to

Table 2: Values of $f(v)$ for the graph in Figure 6

$$\begin{aligned}
f(v_i), i \in \{1,2,...,6\} &= \langle 1,\infty,\infty,\infty,0\rangle \\
f(v_7) &= \langle 3,2,2,\infty,\infty\rangle \\
f(v_8) = f(v_{10}) &= \langle 2,1,1,\infty,\infty\rangle \\
f(v_9) &= \langle 3,\infty,3,2,2\rangle \\
f(v_{11}) &= \langle 4,3,3,\infty,\infty\rangle \\
f(v_{12}) &= \langle 7,7,7,7,7\rangle \\
f(v_{13}) &= \langle 8,8,7,7,7\rangle
\end{aligned}$$

compute $f(v_{11})$, we need $f(v_5) = f(v_6) = \langle 1,\infty,\infty,\infty,0\rangle$ and $f(v_8) = \langle 2,1,1,\infty,\infty\rangle$. Then we compute

$$\begin{aligned}
f(v_{11})[\text{IN0}] &= 1 + \sum_{x\in C(v)} \min\{f(x)[\text{IN0}], f(x)[\text{IN1}], f(x)[\text{OUTC}], f(x)[\text{OUTUD}]\} \\
&= 1 + f(v_5)[\text{IN0}] + f(v_6)[\text{IN0}] + f(v_8)[\text{OUTC}] \\
&= 1+1+1+1 = 4.
\end{aligned}$$

After $f(r)$ has been computed at the root (in this case the root is v_{13}), we call BEST-AT-ROOT to determine $\gamma^1_m(T)$. In this case, $\gamma^1_m(T) = f(v_{13})[\text{OUTC}] = 7$. It is straightforward to backtrack through the tree to determine which vertices comprise a minimum 1-movable dominating set. In this case, since we used $f(v_{13})[\text{OUTC}]$, we know $v_{13} \notin S$. We know $f(v_{13})[\text{OUTC}] = 7$ because $f(v_{12})[\text{IN0}] = 7$, so this means $v_{12} \in S$. Next we examine how $f(v_{12})[\text{IN0}]$ achieves the value 7. In this way, we can determine a 1-movable dominating set S for T satisfying $|S| = \gamma^1_m(T)$. One such choice is $S = \{v_1, v_6, v_7, v_8, v_{10}, v_{11}, v_{12}\}$.

6 Concluding Remarks

Definitions 2.1 and 2.2 describe a variant of domination in which a single sensor can be moved so that the result is also a dominating set. A natural generalization of this idea follows.

Definition 6.1 *Let $k \leq n$ be a natural number. A dominating set $S \subseteq V(G)$ is a* k-movable dominating set *if for every subset $X \subseteq S$ with $X = \{x_1, x_2, \cdots, x_k\}$ there exists a set $Y = \{y_1, y_2, \cdots, y_k\}$ so that $(S - X) \cup Y$ is a dominating set for G and either*

1. y_i is NULL, *or*

2. $y_i \in (V(G) - S) \cap N(x_i)$.

Informally, Definition 6.1 requires that for every size k subset $X \subseteq S$, the sensors located at the vertices in X can be either removed completely (condition 1) or moved to a neighboring vertex that is not in S so that the result, after all of the removals and moves, is again a dominating set for G. This definition is analogous to Definition 2.2. We define the *k-movable domination number* $\gamma_m^k(G)$ as minimum size of a k-movable dominating set in G. Note that if $k = 0$, Definition 6.1 collapses to Dominating Set; hence $\gamma_m^0(G) = \gamma(G)$.

It is interesting to observe that $\gamma_m^k(G)$ (for $k > 1$) and $\gamma_m^1(G)$ are incomparable in the sense that there are graphs for which $\gamma_m^k(G) > \gamma_m^1(G)$ and others for which $\gamma_m^k(G) < \gamma_m^1(G)$. Consider the complete bipartite graph $K_{m,3}$ with $m \geq 3$. Let $\langle A, B \rangle$ be the corresponding bipartition with $|A| = m$ and $|B| = 3$. Any set S that contains one vertex from A and one from B is a 2-movable dominating set. Since no single vertex dominates $K_{m,3}$, we see that $\gamma_m^2(K_{m,3}) = 2$. However, it is easy to see that $\gamma_m^1(K_{m,3}) = 3$. On the other hand, consider the graph G in Figure 7 below. First, observe that $\gamma(G) = 3$ bounds both $\gamma_m^1(G)$ and $\gamma_m^2(G)$ from below. Clearly $\gamma_m^1(G) = 3$; let $S = \{v_{10}, v_{11}, v_{12}\}$. It is not difficult to see that $\gamma_d^2(G) > 3$.

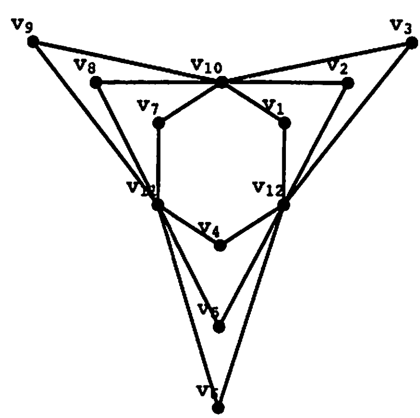

Figure 7: Example where $\gamma_m^2(G) > \gamma_m^1(G)$

It is also interesting to consider other possible generalizations of 1-movable domination. One possibility is to allow a sensor in X to be moved anywhere in its neighborhood, including to another vertex in X. The following alternative generalization of Definition 2.2 formalizes this idea.

Definition 6.2 *Let $k \leq n$ be a natural number. A dominating set $S \subseteq V(G)$ is a* k-movable dominating set (alternative version) *if for every subset $X \subseteq S$ with $X = \{x_1, x_2, \cdots, x_k\}$ there exists a set $Y = \{y_1, y_2, \cdots, y_k\}$ so that $(S - X) \cup Y$ is a dominating set for G and either*

1. *y_i is* NULL, *or*
2. *$y_i \in V(G) \cap N(x_i)$.*

The application of Definition 6.2 to certain graphs can also produce some interesting outcomes. Consider a star graph $K_{1,n-1}$. By Observation 2.4, we know $\gamma_m^1(K_{1,n-1}) = n-1$. Using Definition 6.2, we would find that $\gamma_m^2(K_{1,n-1}) = 2$: take S to be the central vertex of $K_{1,n-1}$ plus any other vertex. Since these two vertices are adjacent, they can swap positions which essentially leaves S unchanged. If on the other hand we use Definition 6.1, $\gamma_m^2(K_{1,n-1}) = n-1$ by taking S to be each leaf vertex.

The definitions given in this section represent two of many possible generalizations of Definition 2.2. It seems a worthwhile direction for further research to consider a specific generalization and then investigate issues related to algorithms, complexity, and approximation for k-movable dominating sets.

References

[1] R. B. BORIE, R. G. PARKER, AND C. A. TOVEY, *Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families*, Algorithmica, 7 (1992), pp. 555–581.

[2] E. COCKAYNE, *Irredundance, secure domination and maximum degree in trees*, Disc. Math., 307 (2007), pp. 12–17.

[3] E. COCKAYNE, O. FAVARON, AND C. MYNHARDT, *Secure domination, weak roman domination and forbidden subgraphs*, Bulletin of the Institute of Combinatorics and its Applications, 39 (2003), pp. 87–100.

[4] M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability*, W. H. Freeman and Co., 1978.

[5] W. GODDARD, S. HEDETNIEMI, S. HEDETNIEMI, *Eternal security in graphs*, J. Combin. Math. Combin. Compute., 52 (2005), pp. 169–180.

[6] S. HEDETNIEMI, *Unsolved algorithmic problems on trees*, AKCE J. Graphs. Combin., 3 (2006), pp. 1–37.

[7] T. WIMER, S. HEDETNIEMI, AND R. LASKAR, *A methodology for constructing linear graph algorithms*, Congressus Numerantium, 50 (1985), pp. 43–60.