# Broadcast Domination

Daniel Lokshtanov

March 8, 2007

## Acknowledgements

First of all, I would like to thank my supervisor Pinar Heggernes. She introduced me to Broadcast Domination and has been a great help and inspiration during the work on this thesis. I always felt welcome in her office, a place where the answers to most of my numerous questions could be found.

I also want to thank the whole Algorithms research group. They welcomed me into the fold the moment I stepped into the faculty building. With their help, I could go on several trips that have played a key role in my academical development.

Finally, thanks to my friends and family. You there for me, whether it is to go for a walk, play soccer, drown my sorrows in wine, make a delicious dinner, watch a movie or simply do nothing.

# Contents

# Chapter 1

# Introduction

The Dominating Set [2, 16] problem is probably one of the most studied problems in graph algorithms. It has been proved $\mathcal{NP}$-complete [6], and so have countless variations of the problem. The Dominating Set problem can be stated as follows: Color the vertices of a graph $G$ black or white so that every white vertex has a black neighbor, using as few black vertices as possible. If we now call the black vertices for *dominators* we see that every dominator dominates its neighbors.

In this thesis, we will look upon a rather recent mutation of the problem statement, called Broadcast Domination. This variant was introduced by Erwin [5] in 2002. In Broadcast Domination, we are free to choose the strength of a dominator. A dominator with *power* $p$ dominates all vertices at distance less than or equal to $p$ from it. Assuming that making a dominator of power $p$ has cost $p$, we can state the Broadcast Domination problem by asking to dominate a graph while minimizing the cost (the sum of powers of the dominators) of doing so.

We observe that if we demand that each dominator has power 1 we return to Dominating Set. Hence in Broadcast Domination we seem to have a much larger search space in which to seek an optimal solution. This naturally leads to the assumption that Broadcast Domination also is $\mathcal{NP}$-complete. Following this assumption, Blair et al. [3] gave polynomial time algorithms for finding an optimal Broadcast Domination in series-parallel graphs, interval graphs and trees. Also, Horton et al. [13] did a computational study of the problem. In this thesis we will show that, quite surprisingly Broadcast Domination is in fact polynomial time solvable on all input graphs, by presenting an $O(n^6)$ algorithm. To do this, we will not apply any of the standard techniques for proving polynomial time solvability, like reduction to 2-SAT or Dynamic Programming. Instead we will reduce the problem of finding an optimal broadcast domination to a set of shortest path problems in auxiliary graphs. At the end of the thesis we will also discuss a couple of variations of the problem.

## 1.1 Notation and terminology

A graph $G = (V, E)$ is a set $V = V(G)$ of *vertices* and a set $E(G) = E \subseteq V^2$ of *edges*. In this thesis, we will say that $G$ has $|V| = n$ vertices and $|E| = m$ edges. If $(v, v) \notin E$, $G$ is referred to as a *simple* graph. If $(u, v) \in E \Rightarrow (v, u) \in E$ the graph is *undirected*, otherwise it is *directed*. If some *weight function* $w : E \rightarrow \mathbb{R}$ is supplied, we say that $G$ is *weighted* and that $w((u, v))$ is the *weight* of the edge $(u, v)$. Similarly, a weight function $w' : V \rightarrow \mathbb{R}$ may be defined on the vertices. All graphs in this thesis are simple and undirected unless stated otherwise. For a simple undirected graph $G$, its *complement*, is defined as $\overline{G} = (V, \{(u, v) : u \neq v, (u, v) \notin E\})$.

A sequence $P$ of vertices, $\{p_1, p_2 ... p_n\}$ where $(p_i, p_i + 1) \in E$ is called a *walk*. $n - 1$ is the *length* of the walk. If $i \neq j \rightarrow p_i \neq p_j$ $P$ is a *path*. A *cycle* is a path of length at least three, with the exception that $p_1 = p_n$. The *distance*, $\delta(u, v)$ between two vertices $u$ and $v$ is the length of the shortest path starting in $u$ and ending in $v$. If we refer to the distance in another graph than $G$ we will use subscript. For instance the distance between $u$ and $v$ in the graph $H$ is denoted $\delta_H(u, v)$. For a vertex $u$ we will call $e(u) = \max_{v \in V}(\delta(u, v))$ the *eccentricity* of $u$. The maximum eccentricity over all vertices of $G$ is the *diameter* of $G$, $diam(G)$, the minimum eccentricity over all vertices is the graph's *radius*, $rad(G)$.

A graph $H = (V_H, E_H)$ is a *subgraph* of $G = (V, E)$ if $V_H \subseteq V$ and $E_H \subseteq E$. Given a subset of the vertices, $W \subseteq V$, the *subgraph induced by* $W$ is the graph $G[W] = (W, \{(u, v) : (u, v) \in E \wedge (u, v) \in W^2\})$

For a vertex $v$ and natural number $r$, the *ball* with *center* in $v$, and *radius* $r$ is the set of vertices $B(v, r) = \{u : \delta(u, v) \leq r\}$. The *closed neighborhood* of a vertex $v$, $N[v]$, is simply the ball $B(v, 1)$. The *neighbors* of $v$ is the set $N(v) = N[v] \setminus v$. For a set of vertices $S$, define $N[S] = \bigcup_{u \in S} N[u]$ and $N(S) = N[S] \setminus S$. The *degree* of a vertex $v$ is $d(v) = |N(v)|$. The *maximum degree* of a graph $G$ is $\Delta(G) = \max_{v \in V} d(v)$.

A *clique* $K$ in a graph $G$ is a subset of the vertices of $G$ so that for each distinct $u$ and $v$ in $K$, $(u, v) \in E(G)$. An *independent set* $I$ is a clique in $\overline{G}$. A set $D$ of vertices is a dominating set in $G$ if $N[D] = V(G)$. A function $f : V \rightarrow \mathbb{Z}^+$ is a *broadcast*. A broadcast is a *broadcast domination* if for every vertex $u$ there is a vertex $v$ with $\delta(u, v) \leq f(v)$. Given a broadcast $f$, the *cost* incurred by a vertex set $S \subseteq V$ is $C_f(S) \sum_{v \in S} f(v)$. The Optimal Broadcast Domination problem asks to find a broadcast domination $f$ of $G$ that minimizes $C_f(V)$. We will call such a broadcast domination an *optimal broadcast domination*. If a simple directed graph $D = (V, E)$ has no cycles, we say that $D$ is *acyclic*. A *topological ordering* of the directed graph $D$ is an ordering of the vertices $(v_1, v_2\ , \ ...\ , v_n)$ so that for every edge $(v_i, v_j)$ in $E$ we have that $i < j$. One can show that a simple directed graph $D$ has a topological ordering if and only if it is acyclic [14].

A *graph class* $\mathcal{G}$ is a (possibly infinite) set of graphs. A graph $G \in \mathcal{G}$ is said to *it belong* to the graph class, or to be a *member* of the graph class. Two commonly used graph classes are the *planar graphs* - all graphs that can be embedded in a plane without crossing edges and *trees* - connected graphs with no cycles.

As a rule of citation, all results that are not new in this thesis will be cited. The only exceptions from this rule are the results we consider to be a part of the folklore.

## 1.2 Outline

In chapter 2 we give a brief introduction to complexity theory. We cover asymptotic notation, the classes $\mathcal{P}$ and $\mathcal{NP}$, and introduce $\mathcal{NP}$-completeness. In chapter 3 we consider the Dominating Set problem, along with a few similar problems. We define Optimal Broadcast Domination and discuss the relation between Optimal Broadcast Domination and Dominating Set. Chapter 4 covers structural results on broadcast dominations. The notions of efficiency and domination graphs are introduced and used to provide the tools needed to devise a polynomial time algorithm for Optimal Broadcast Domination. The algorithm itself, together with a correctness proof and a complexity analysis is presented in chapter 5. In chapter 6 we cover a couple generalizations of Optimal Broadcast Domination, while chapter 7 summarizes and concludes the results of the thesis.

# Chapter 2

# Some complexity theory

In this chapter, we will review the classes $\mathcal{P}$ and $\mathcal{NP}$ and give an explanation of what $\mathcal{NP}$-complete problems are. For the purpose of this thesis one only needs to understand the basic ideas of these concepts, so the explanations are slightly simplified. For a more complete and rigorous coverage of the topics, please refer to [17] or [14].

## 2.1 Big-Oh notation and the class $\mathcal{P}$

When analyzing the efficiency of an algorithm we usually want to study how it behaves on large input instances. Also, experience has shown that the average running time of an algorithm is usually close to the worst case running time. Hence the classical performance analysis relies on giving a function $f$ expressing the number of steps performed by the algorithm on the worst possible input of size $n$. However, this function $f$ tends to be complicated, and to have many lower order terms that are negligible on large enough inputs. To avoid finding and writing all these terms we apply $\mathcal{O}$-notation (read "Big-Oh") that suppresses all lower order terms and constant factors. This means that $10n^4 + 3n^3 + n\log(n) = \mathcal{O}(n^4)$ and $100 \times 2^n + 200 \times 3^n = \mathcal{O}(3^n)$.

A natural question to ask, is for which problems we have efficient algorithms. By efficient, we mean that we are able to apply the algorithm on large input instances, and get an answer in reasonable time. Now, this is not a precise definition, so it is fairly hard to work with. Therefore we say that an algorithm is efficient if it runs in polynomial time. That is, if there exists a constant $c$ independent of the input so that the algorithm runs in $\mathcal{O}(n^c)$ time. Now, obviously this definition is to wide - by this definition an algorithm running in $\mathcal{O}(n^{100})$ is efficient, something that clearly is not the case. In fact, if executed today on the earth's fastest computer, the algorithm would not terminate within the life span of the universe even on an input of size 2! However, incomplete as this definition may seem, it has proven practical. This is because most problems that we find polynomial time algorithms for are fast, that is the exponent $c$ mentioned above is 10 or less. We call the set of all problems that are solvable in polynomial time $\mathcal{P}$.

## 2.2 Reductions

Unfortunately, many important problems have proven very hard to find efficient algorithms for. These problems are often easy to formulate, but somehow, in their simplicity they seem not to be solvable by anything but a brute force search. Finding a clique or an independent set of a given size in a graph are classical examples of such problems. We will refer to these problems as Clique and Independent Set respectively. Also, assigning values to boolean variables in order to satisfy a given boolean formula is another. In 1971/72 Cook and Levin provided an explanation for why these problems are so difficult. They showed that all these problems are interconnected, and that by solving one of them, we in fact solve them all.

In order to do this, they use the concept of *reductions* from one problem to another. A reduction is simply an algorithm to transform an instance of one problem to an instance of another, so that the solution of the second can be used to solve the first. For decision problems - problems where the output should be either yes or no, we formalize this notion. For a decision problem $P$ and an instance $I$ of the problem, let the answer to $I$ with respect to $P$ be denoted $A_P(I)$. This answer is either $True$ or $False$. Now, a reduction from a problem $P$ to another problem $P'$ is a function $f$ that takes instances of $P$ to instances of $P'$ so that $A_{P'}(f(I)) = A_P(I)$. We say that $f$ is a *polynomial time reduction* if there is a polynomial time algorithm computing $f$.

A simple example of a polynomial time reduction is one from Clique to Independent Set. We simply let $f(G, k) = (\overline{G}, k)$. By the definition of an independent set we see that $G$ has a clique of size $k$ if and only if $\overline{G}$ has an independent set of size $k$. So, what does the existence of this reduction tell us? Well, suppose you had a polynomial time algorithm that solves Independent Set. Then, in order to solve any instance $I$ of Clique, we could simply apply the reduction above, and run the polynomial time algorithm on $f(I)$. Now, since $f$ is a reduction, the answer given by the Independent Set algorithm in fact also is the answer to our instance of Clique. Hence, any polynomial time algorithm for Independent Set can be transformed into a polynomial time algorithm for Clique. Obviously this example can be generalized. That is, if there is a polynomial time reduction from some problem $A$ to a problem $B$ then a polynomial time algorithm for $B$ automatically yields a polynomial time algorithm for $A$. Seen from this point of view we can say that $B$ is at least as hard as $A$, or $A \leq_T B$.

## 2.3 The class $\mathcal{NP}$ and $\mathcal{NP}$-completeness

A property shared by the problems mentioned above is that they are all *polynomial time verifiable*, that is, if we are given a problem instance and a solution, there is a polynomial time algorithm that can check whether the proposed solution indeed is a solution of the given instance. For instance, given a graph $G$ and a clique $K$, it is easy to check whether $K$ actually is a clique in $G$. Similarly to $\mathcal{P}$, we define $\mathcal{NP}$ to be the class of all polynomial time verifiable problems.

What Cook and Levin realized is that some of the problems in $\mathcal{NP}$ are special in the way that they can be used to encode all other problems in the class. We say that a problem $A$ is $\mathcal{NP}$-complete, or belongs to the class $\mathcal{NPC}$ if $A \in \mathcal{NP}$ and for all $B \in \mathcal{NP}$, $B \leq_T A$. The famous result of Cook and Levin states that 3-SAT is

$\mathcal{NP}$-complete. The formal definition of 3-SAT is out of the scope of this thesis. If interested, the reader may refer to [17] for this definition, and for the proof of the following theorem.

**Theorem 2.3.1** *3-SAT* $\in \mathcal{NPC}$.

So, what are the consequences of this result? Well, suppose we find a polynomial time algorithm for 3-SAT. Then, by definition of $\mathcal{NP}$-completeness we automatically have polynomial time algorithms for all problems in $\mathcal{NP}$. However, from an intuitive standpoint it seems farfetched that all polynomial time verifiable problems also are polynomial time solvable. After all, it is easier to check a solution for correctness than actually finding it. Still, as of today, one has neither been able to find a polynomial time algorithm for 3-SAT, nor prove that none exists. In fact, this problem is known as the famous open $\mathcal{P} =^? \mathcal{NP}$ problem.

3-SAT is not the only problem to have this property. For instance, it is a simple exercise to show that 3-SAT $\leq_T$ Clique [17] which proves that Clique is in $\mathcal{NPC}$. As we have already showed that Clique $\leq_T$ Independent Set, so is Independent Set. Since Cook and Levin published their result, hundreds of other problems have been proved $\mathcal{NP}$-complete. Garey and Johnson [6] provides a list of over 300 of them.

To conclude this chapter we will give a proof that Dominating Set is $\mathcal{NP}$-complete by presenting a reduction from Independent Set. Of course, the problem has been shown $\mathcal{NP}$-complete a long time ago, yet we chose to present the proof as this problem has a key role in the thesis and we wish to give an example of a nontrivial reduction. As we mentioned in the introduction the problem of Dominating Set asks as to find a minimum cardinality dominating set in a given graph. This minimum cardinality will be denoted $\gamma(G)$.

For a given instance $G = (V, E)$ of Independent Set we build a graph $G' = (V', E')$ $= ((V \cup E), E \cup \{(u, (u, v)) : u \in V, (u, v) \in E\})$. To complete the reduction we need a result concerning the dominating sets of $G'$.

**Lemma 2.3.2** *For every dominating set $D$ of $G'$ with $D \setminus V \neq \emptyset$ there is a dominating set $D'$ of $G'$ with $|D'| \leq |D|$ and $D' \setminus V \subset D \setminus V$.*

**Proof.** Suppose that $D \setminus V \neq \emptyset$. Then there is a $(u, v) \in D \setminus V$. Now, consider $D' = (D \cup \{u\}) \setminus \{(u, v)\}$. Obviously $|D'| \leq |D|$ as $(u, v)$ is removed. Also $D'$ is a dominating set as $N[(u, v)] \subseteq N[u]$. Finally $D' \setminus V \subset D \setminus V$ as $(u, v) \notin V$ and $u \in V$. ∎

**Corollary 2.3.3** *For every dominating set $D$ of $G'$ with $D \setminus V \neq \emptyset$ there is a dominating set $D'$ of $G'$ with $|D'| \leq D$ and $D' \setminus V = \emptyset$.*

Using this, we can derive a result connecting the independent sets of $G$ to dominating sets of $G'$.

**Theorem 2.3.4** *$G$ has an independent set of size at least $k \Leftrightarrow G'$ has a dominating set of size at most $|V| - k$.*

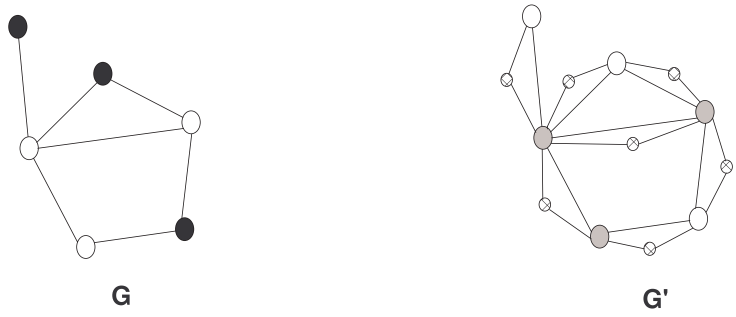


Figure 2.1: *On the left hand side, the black nodes indicate the independent set. On the right hand side, small nodes are the nodes added to $G$ to construct $G'$. The grey nodes indicate a minimum dominating set of $G'$*

**Proof.** Suppose $V$ has an independent set $I$ of size at least $k$. Then $D = V \setminus I$ has size at most $|V| - k$. Also, for all $(u, v) \in E$ either $u \in D$ or $v \in D$. Hence, $D$ is a dominating set of $G'$. In the other direction, suppose $G'$ has a dominating set $D$ of size at most $|V| - k$. By Corollary 2.3.3 there is another dominating set $D' \subseteq V$ with size at most $|V| - k$. Consider $I = V \setminus D'$. First, $I$ has size at least $|V| - (|V| - k) = k$. Additionally for all $(u, v) \in E$ either $u \notin I$ or $v \notin I$. This yields that $I$ is an independent set of $G$ completing the proof ∎

**Corollary 2.3.5 (Folklore)** *Independent Set $\leq_T$ Dominating Set, and Dominating Set is $\mathcal{NP}$-complete*

In fact, Dominating Set is such a difficult problem that it remains $\mathcal{NP}$-complete even when input is restricted to specific graph classes. In this thesis, we will use the fact that Dominating Set is in $\mathcal{NPC}$ even if input is restricted to the planar graphs.

**Theorem 2.3.6** *[6] Planar Dominating Set is $\mathcal{NP}$-complete*

# Chapter 3

# Dominating Set and some of its variants

The Dominating Set problem was introduced in the late 1950's [2, 16]. The problem arises naturally from many different situations. For instance, consider the problem of deciding where to build fire stations in a city. One would want every house in the city to be within a certain range from a fire station. At the same time each fire station costs time and money to operate, thus we wish to minimize the number of fire stations to be built. We model this as a graph problem. Let the locations in the city be the vertices of a graph $G = (V, E)$. Now put an edge between two locations if the fire-truck can get between them in time. It is easy to see that a minimum cardinality dominating set $S$ in $G$ corresponds to an optimal placement of the fire stations.

## 3.1 Power Domination

Dominating Set also arises from the need to monitor electric networks. Consider a network having components and links between these components. A measurement unit can be attached to any one of these components and measure the state of the component it is attached to and all its neighbors. If we wish to monitor all components of the network while minimizing the number of measurement units used, again we arrive at the Dominating Set problem. In practice, however, a measurement unit can sometimes monitor components outside of the units neighborhood. This can only happen if certain conditions are satisfied, but the difference from Dominating Set is significant enough for an alternate model to be used. This model leads to the problem of Power Domination, one of the many spin-off problems from Dominating Set. More detailed coverage of Power Domination can be found in [8, 7].

## 3.2 Broadcast Domination

Another way to view Dominating Set achieved through is the problem of placing radio transmitters. Given a graph $G$ representing cities, we wish to place as few radio transmitters as possible on the map so that every city can hear a transmitter. A city can hear a transmitter placed in the city itself or in any of its neighbors. This model, however, does not capture the fact that some radio transmitters are stronger than others. Broadcast Domination, a model taking this into account, was introduced in 2002 by Erwin [5]. In this scenario we are given an input graph $G = (V, E)$ and are to produce an integer valued function $f : V \to Z^+$ describing the placement and strength of the different transmitters. The transmitters are placed at the vertices with a nonzero transmission power. That is, we use $f$ to select the *dominators*, $V_f = \{v \in V : f(v) \geq 1\}$, and the other vertices, called *receivers*. We say that each dominator $v \in V_f$ has *power* $f(v)$. Now, as with Dominating Set we want all vertices of $G$ to be dominated by some dominator. Formally we say that $f$ is a *broadcast domination* if for every vertex $u$ there is a dominator $v \in V_f$ satisfying $\delta(u, v) \leq f(v)$. In this case we say that the vertex $u$ *hears* vertex $v$. If $|V_f| = 1$ we say that $f$ is *radial*.

When dealing with Dominating Set we wish to minimize the number of dominators. Similarly, in Optimal Broadcast Domination we wish to minimize the cost of dominating all the vertices. We define the *cost* incurred by a set $S \subseteq V$ of vertices to be $C_f(S) = \sum_{v \in S} f(v)$. Thus, the Optimal Broadcast Domination problem asks to find a broadcast domination $f$ minimizing $C_f(V)$.

## 3.3 $r$-Domination and the $(k, r)$-Center Problem

If we demand that all dominators are to have a certain power $r$, we arrive at $r$-Domination [11, 18]. This is clearly a $\mathcal{NP}$-complete problem, as setting $r$ to one returns us to the original Dominating Set problem. Another problem similar to $r$-Domination is the $(k, r)$-Center problem in which we seek to find an $r$-Domination of $G$ using $k$ vertices. One of these parameters is given, the objective is to minimize the other [1, 6]. If the given parameter is $r$ this is exactly $r$-domination, hence this variant of Dominating Set is $\mathcal{NP}$-complete as well.

# Chapter 4

# Structural results on Broadcast Domination

Intuitively Optimal Broadcast Domination looks harder than Dominating Set, because an algorithm solving the problem not only has to determine which vertices are to be dominators, but also what power each dominator is to have. However, a second glance reveals a flaw in this logic. For instance, from a basic result in probabilistic graph theory we know that " “almost all” graphs have radius 2. This means that as $n$ tends to infinity the probability that a randomly drawn graph on $n$ vertices has radius 2 tends to 1. As a consequence, one will " “almost always” be able to find a radial broadcast domination of cost 2 in large graphs. Thus, one can expect that the set of optimal solutions in Optimal Broadcast Domination is a lot less diverse than that of Dominating Set. In fact, the polynomial time algorithm we present will exploit that we can always find an optimal broadcast domination having specific structural properties.

## 4.1 Lower and upper bounds

Before we start working on an algorithm to find an optimal broadcast domination, we would like to have some simple bounds on the cost of an optimal broadcast domination in a given graph. For a given graph, we will denote this cost by $\gamma_b(G)$. First of all, we notice that if there is a vertex $v$ with $f(v) \geq diam(G)$ then $f$ is a broadcast domination. Thus, it is never necessary to consider broadcast dominations with powers greater than $diam(G)$. Following this observation, in the literature broadcast dominations are often defined as functions from $V$ to $\{0\ ,...,\ diam(G)\}$ instead of from $V$ to the nonnegative integers. We observe that $\gamma_b(G) \leq rad(G)$ because we can dominate the entire graph by finding a vertex $v$ with eccentricity $rad(G)$ and make a radial broadcast domination $f$ with $f(v) = rad(G)$.

Proving a lower bound is slightly harder.

**Lemma 4.1.1** *Let $P = \{u_1, u_2 .... u_k\}$ be a shortest path from $u_1$ to $u_k$. Then, for any vertex $u$ and positive integer $p$, $|B(u,p) \cap P| \leq 2p + 1$.*

**Proof.** Suppose for contradiction that $|B(u,p) \cap P| > 2p+1$. Then there must be two vertices $v$ and $w$ on $P$ with $\delta_P(v,w) > 2p$. But $P$ is a shortest path, so $\delta_P(v,w) =$

$\delta(v,w)$. This gives us $\delta(v,w) > 2p = \delta(v,u) + \delta(u,w) \geq \delta(v,w)$, a contradiction. ■

Using this lemma we can prove a simple lower bound on $\gamma_b(G)$

**Lemma 4.1.2** $\frac{diam(G)}{3} \leq \gamma_b G$

**Proof.** Let $f$ be an optimal broadcast domination, Let $x$ be a vertex with eccentricity $diam(G)$ and let $y$ be a vertex with $\delta(x,y) = diam(G)$. Let $P$ be a shortest path from $x$ to $y$. Now, $\sum_{v \in V_f} |B(v, f(v) \cap P| \geq |P| = diam(G)$ as $f$ is a broadcast domination. As $P$ is a shortest path we have that $\sum_{v \in V_f} |B(v, f(v) \cap P| \leq \sum_{v \in V_f}(2f(v)+1) = |V_f| + 2\gamma_b(G) \leq 3\gamma_b(G)$. Together these inequalities give $diam(G) \leq 3\gamma_b(G)$. ■

## 4.2 Balls, Efficiency and Optimal Substructure

Given a graph $G$, for a vertex $v$ and positive integer $r$ we defined the *ball centered at* $v$ *with radius* $r$, to be $B(v,r) = \{u : \delta(u,v) \leq r\}$. Notice, that for a broadcast domination $f$, each ball $B(v, f(v))$ for $v \in V_f$ corresponds to a transmitter and all vertices that can hear it. As a slight abuse of notation, when we talk about a ball, we might also mean the subgraph of $G$ induced by the ball. Finally, note that a function $f$ is a broadcast domination if and only if $\bigcup_{v \in V_f} B(v, f(v)) = V$.

Now we are ready to state the first structural result about optimal broadcast dominations. It says that there is an optimal broadcast domination in which every vertex hears exactly one transmitter. We call such a broadcast domination *efficient*. The Lemma is due to Dunbar et al. [4].

The proof idea is as follows - if there is a vertex hearing two transmitters, the two transmitters are fairly close to each other. Then we can remove the two transmitters and replace them by one strong transmitter in the middle.

**Lemma 4.2.1** *[4] For any inefficient broadcast domination $f$ of $G$ there is an efficient broadcast domination $f'$ with $C_{f'}(V) \leq C_f(V)$ and $|V_{f'} < V(f)|$.*

**Proof.** Suppose that we have an inefficient broadcast domination $f$. Then there is a vertex $u$ hearing both $v$ and $w$ in $V_f$. Then we know that $\delta(v,w) \leq \delta(v,u) + delta(u,w) \leq f(v) + f(w)$. Hence, on the shortest path from $v$ to $w$ there is a vertex $x$ satisfying $\delta(v,x) \leq f(w)$ and $\delta(w,x) \leq f(v)$. Now, we let $f'(y) = f(y)$ for $y \in V \setminus v, w, x$, $f'(v) = f'(w) = 0$ and $f'(x) = f(v) + f(w)$. Obviously $|V_{f'}| < |V_f|$ and $C_{f'}(V) = C_f(V)$. We only need to prove that $f'$ is indeed a broadcast domination. But we know that for any $y \in B(v, f(v))$ we have that $\delta(y,x) \leq \delta(y,v) + \delta(v,x) \leq f(v) + f(w) \leq f'(x)$ so $y \in B(x, f'(x))$. Similarly any $y$ in $B(w, f(w))$ is contained in $B(x, f'(x))$, implying $B(v, f(v)) \cup B(w, f(w)) \subseteq B(x, f'(x))$. Finally, we know that $\bigcup_{u \in V_{f'}} B(u, f'(u)) = \bigcup_{u \in V_f} B(u, f(u)) \setminus (B(v, f(v)) \cup B(w, f(w))) \cup B(x, f'(x)) = V$ proving that $f'$ is a broadcast domination. ■

**Corollary 4.2.2** *[4] For every graph there is an optimal efficient broadcast domination*

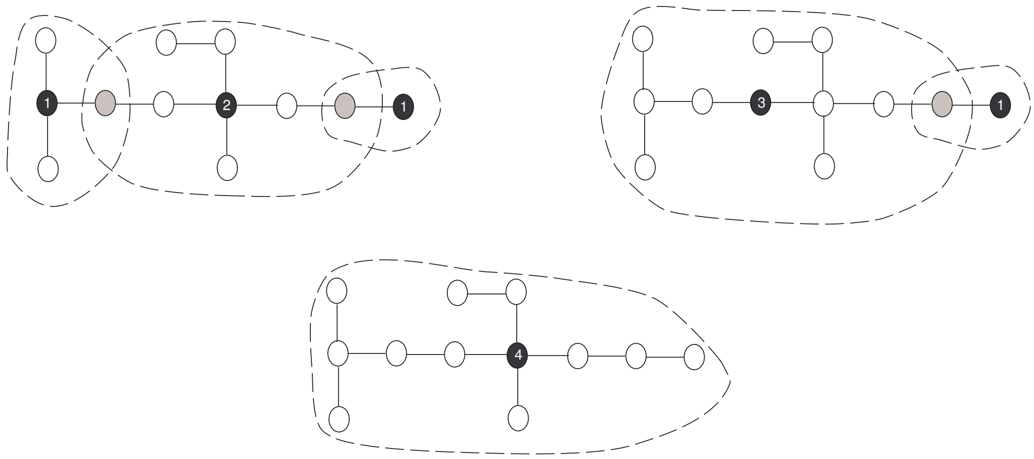


Figure 4.1: *Converting an optimal inefficient broadcast domination into an efficient one. Black nodes are dominators. Numbers in the nodes indicate transmission power. The dashed lines indicate which receivers hear a transmitter. Grey nodes are receivers that hear more than one transmitter.*

**Proof.** Apply Lemma 4.2.1 and induction ∎

To conclude the section, we note that we can apply a cut and paste argument to show that Optimal Broadcast Domination has optimal substructure.

**Lemma 4.2.3** *Let $f$ be an optimal broadcast domination of $G$ and $S \subseteq V_f$, Now let $V' = \bigcup_{v \in S} B(v, f(v))$. Then $f' : V' \to N$ defined to be $f'(v) = f(v)$ is an optimal broadcast domination of $G[V']$.*

**Proof.** Suppose for contradiction that $f$ is an optimal broadcast domination of $G$ but that $f'$ is not optimal in $G[V']$. Then there is a broadcast domination $g'$ of $G[V']$ with $C_{g'}(V') < C_{f'}(V')$. Now, let $g(v) = g'(v)$ for v in $V'$ and $g(v) = f(v)$ for $v \in V \setminus v'$. Obviously, $g$ is a broadcast domination of $G$. Also $C_g(V) = C_g(V') + C_g(V \setminus V') < C'_f(V') + C_g(V \setminus V') = C_f(V') + C_f(V \setminus V') = C_f(V)$, a contradiction as $f$ is optimal. ∎

## 4.3 Domination Graphs

A *domination graph* is a structure that visualizes how a given efficient broadcast domination dominates a graph. The purpose of a domination graph is to show the strength of the different transmitters, and how the transmitters relate to each other. Formally, given a graph $G = (V, E)$ and an efficient broadcast domination $f$, the domination graph is a graph $G_f = (V_f, E_f)$. The vertex set of $G_f$ is just the set of dominators. For two vertices $u$ and $v$ in $V_f$ there is an edge $(u, v)$ in $E_f$ if there is an edge in $G$ between a vertex in $B(u, f(u))$ and a vertex in $B(v, f(v))$. Observe that one can obtain the domination graph from the original one by contracting all receivers into the dominator they hear.

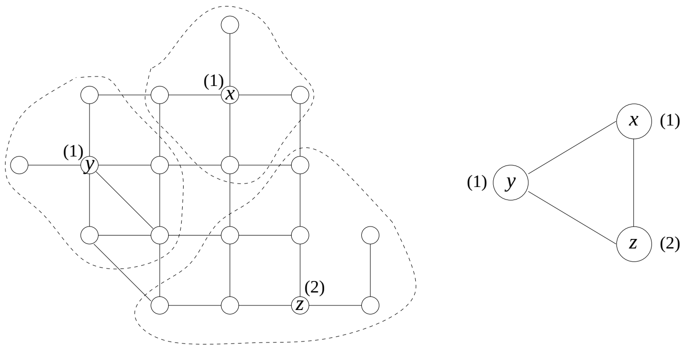


Figure 4.2: *On the left hand side, a graph $G$ with an efficient broadcast domination $f$ is shown. For vertices $v$ with $f(v) \geq 1$, the broadcast powers $f(v)$ are shown in parentheses, and the dashed curves indicate the balls $B(v, f(v))$. For all other vertices $w$, $f(w) = 0$. On the right hand side, the corresponding domination graph $G_f$ is given, and the weight of each vertex is shown in parentheses.*

Observe that if $u$ and $v$ are neighbors in $G_f$, then there is a vertex $x$ in $B(v, f(v))$ and $y$ in $B(u, f(u)$ so that $(u, v) \in E$. Hence $\delta(u, v) \leq f(u) + f(v) + 1$. As $f$ is efficient $\delta(u, v) > f(u) + f(v)$. We conclude that $\delta(u, v) = f(u) + f(v) + 1$.

## 4.4 The structure of an optimal broadcast domination

When we showed that there is an optimal broadcast domination that is efficient, we did it by showing that if two dominators were close enough we could remove them and replace them with one strong. If we inquire whether a similar technique can be applied in other situations we see that if you have a domination graph with a vertex of degree 3 or more, the vertex and its neighbors can be replaced by a single dominator. By repeatedly applying this technique we get an efficient broadcast domination whose domination graph has no vertices of degree larger than 2, that is $G_f$ is a path or a cycle.

**Lemma 4.4.1** *Let $f$ be an efficient broadcast domination on $G = (V, E)$. If the domination graph $G_f$ has a vertex of degree $> 2$, then there is an efficient broadcast domination $f'$ on $G$ such that $|V_{f'}| < |V_f|$ and $c_{f'}(V) = c_f(V)$.*

**Proof.** Let $v$ be a vertex with degree larger than 2 in $G_f$, and let $x, y$, and $z$ be three of the neighbors of $v$ in $G_f$. By the way the domination graph $G_f$ is defined, $v, x, y$, and $z$ are also vertices in $G$, and they all have broadcast powers $\geq 1$ in $f$. Since $f$ is efficient, $\delta(v, x) = f(v) + f(x) + 1$. Similarly, $\delta(v, y) = f(v) + f(y) + 1$ and $\delta(v, z) = f(v) + f(z) + 1$. Assume without loss of generality that $f(x) \leq f(y) \leq f(z)$.

If $f(x) + f(y) > f(z)$ then we construct a new broadcast $f'$ on $G$ with $f'(u) = f(u)$ for all vertices $u \in V \setminus \{v, x, y, z\}$. Furthermore, we let $f'(v) = f(v) + f(x) + f(y) + f(z)$, and $f'(x) = f'(y) = f'(z) = 0$. The new broadcast $f'$ is dominating since every vertex that was previously dominated by one of $v, x, y$, or $z$ is now dominated by $v$. To see this, let $u$ be any vertex that was dominated by $x, y$, or $z$ in $f$. Thus $\delta_G(v, u) \leq f(v) + 2f(z) + 1$ by our assumptions. Since $f'(v) > f(v) + 2f(z)$, vertex $u$ is now dominated by $v$ in $f'$. The cost of $f'$ is the same as that of $f$, and the number of dominators in $f'$ is smaller.

Let now $f(x) + f(y) \leq f(z)$. As we mentioned above, there is a path $P$ in $G$ between $v$ and $z$ of length $f(v)+f(z)+1$. Let $w$ be a vertex on $P$ such that $\delta_P(w, z) = f(v) + f(x) + f(y)$. Since $f$ is efficient, $f(w) = 0$. We construct a new broadcast $f'$ on $G$ such that $f'(u) = f(u)$ for all vertices $u \in V \setminus \{v, w, x, y, z\}$. Furthermore, we let $f'(w) = f(v) + f(x) + f(y) + f(z)$, and $f'(v) = f'(x) = f'(y) = f'(z) = 0$. By the way $\delta_G(z, w)$ is defined, any vertex that was dominated by $z$ or $v$ in $f$ is now dominated by $w$, since $\delta_G(v, w) < f(z)$. Let $u$ be a vertex that was dominated by $y$ in $f$. The distance between $u$ and $w$ in $G$ is $\leq 2f(y) + 2f(v) + f(z) + 2 - f(v) - f(x) - f(y) = f(y) + f(v) + f(z) + 2 - f(x) \leq f(y) + f(v) + f(z) + f(x) = f'(w)$. Thus $u$ is now dominated by $w$. The same is true for any vertex that was dominated by $x$ in $f$ since we assumed that $f(x) \leq f(y)$. Thus $f'$ is a broadcast domination. Clearly, the costs of $f'$ and $f$ are the same, and $f'$ has fewer broadcast dominators.

If $f'$ is efficient, we have a new broadcast domination as desired, and the lemma follows. If $f'$ is not efficient, then by Lemma 4.2.1 there exists an efficient broadcast domination with the same cost as that of $f'$ and with fewer broadcast dominators than those of $f'$. The proof of Lemma 4.2.1 is constructive, so we can apply it to modify $f'$ as desired, and the proof is complete. ∎

Using this lemma we can formalize the discussion at the beginning of this section.

**Theorem 4.4.2** *For any graph $G$, there is an efficient optimal broadcast domination $f$ on $G$ such that the domination graph $G_f$ is either a path or a cycle.*

**Proof.** Let $f$ be any efficient optimal broadcast domination on $G = (V, E)$. If $G_f$ has a vertex of degree larger than 2 then by Lemma 4.4.1, there is an efficient broadcast domination $f'$ on $G$ with $|V_{f'}| < |V_f|$ and $c_{f'}(V) = c_f(V)$. We proceed by downwards induction until the base case $|V_f| = 1$. In this case the domination graph only has one vertex thus having no vertices of degree larger than 2. As a domination graph of a connected graph is connected it must be either a path or a cycle and the theorem follows. ∎

Having shown the main structural result of this chapter, we proceed to proving a couple of corollaries that will prove useful to deduce correctness of the algorithm. Also, for technical reasons we will define the *path broadcast domination number*, $\gamma_{bp}(G)$ to be the minimum cost over all broadcast dominations of $G$ with paths as their domination graph. Notice that the upper and lower bounds we showed for $\gamma_b(G)$ also hold for $\gamma_{bp}(G)$. Thus we end up with the following chain of inequalities; $\frac{diam(G)}{3} \leq \gamma_b(G) \leq \gamma_{bp}(G) \leq rad(G)$.

**Corollary 4.4.3** *For any graph $G = (V, E)$, there is an efficient optimal broadcast domination $f$ on $G$ such that removing the vertices of $B(v, f(v))$ from $G$ results in at most two connected components, for every $v \in V_f$.*

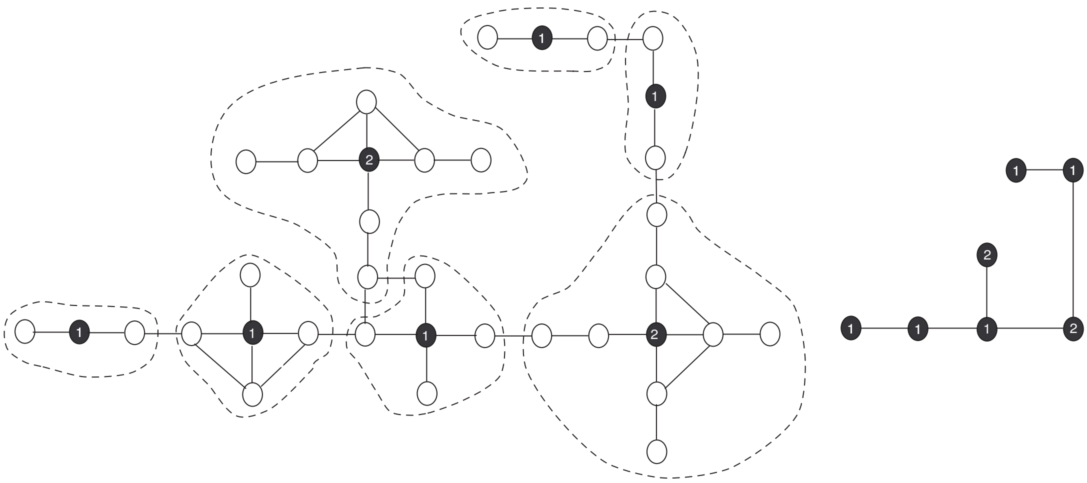


Figure 4.3: *An optimal efficient broadcast domination with a vertex of degree* 3 *in the domination graph. The domination graph is shown on the right.*

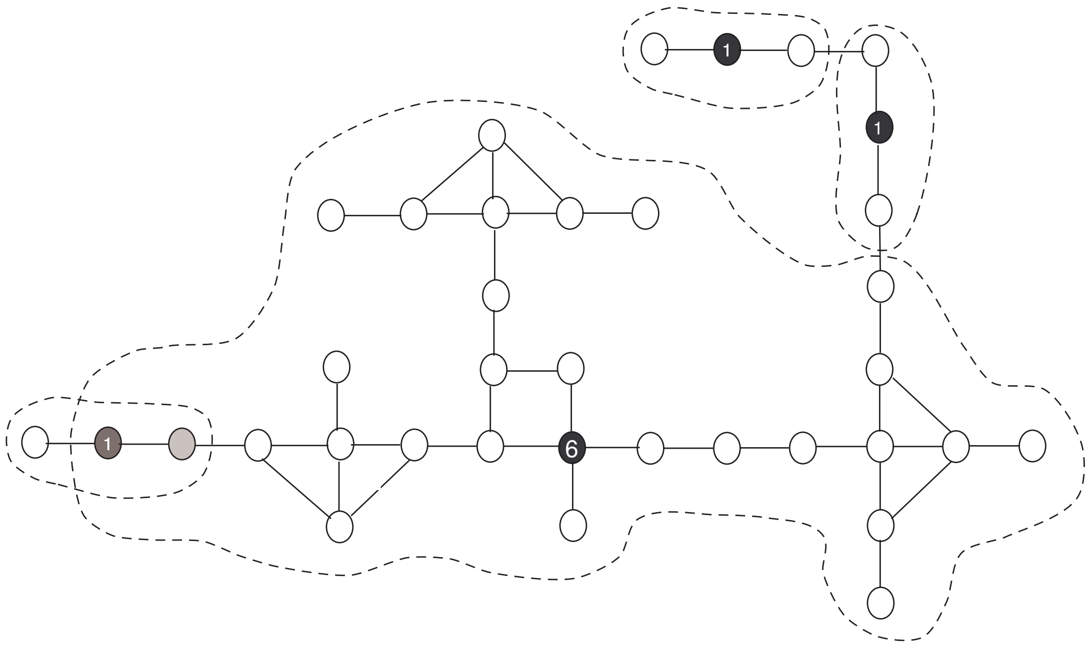


Figure 4.4: *The dominator with degree* 3 *in the domination graph and its neighbors have been replaced by a strong dominator. Now the broadcast domination is not efficient*

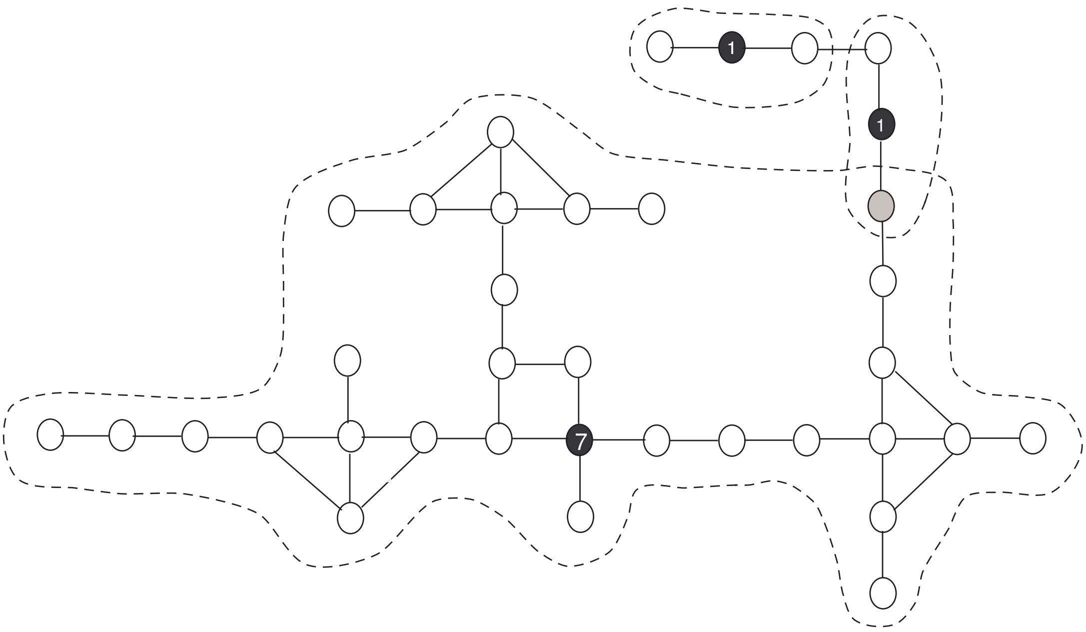


Figure 4.5: *After yet a replacement the broadcast domination is still not efficient*

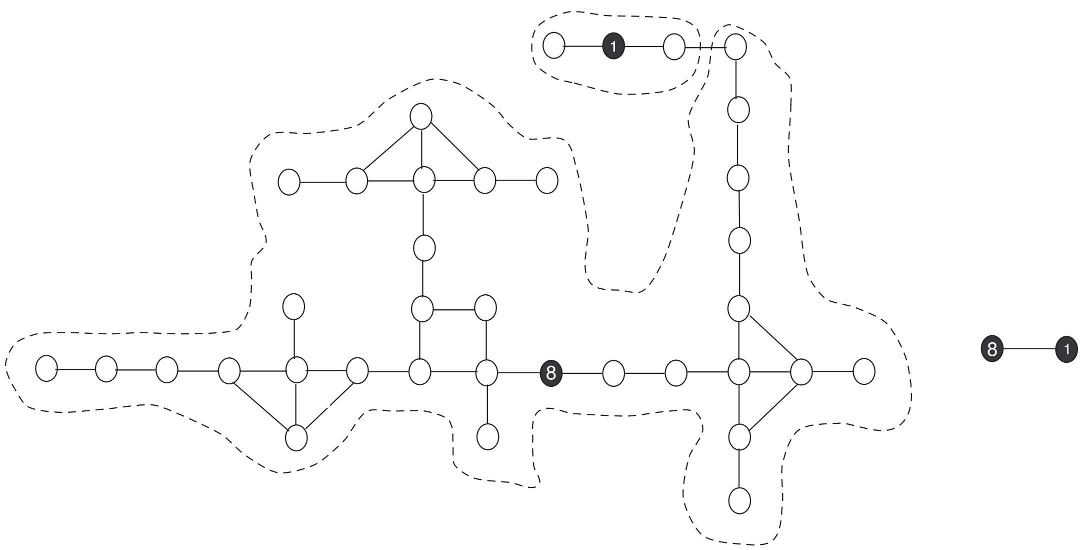


Figure 4.6: *The new broadcast domination is efficient and has a path as its domination graph*

**Proof.** Since there is always an efficient optimal broadcast domination $f$ on $G$ such that the balls $B(v, f(v))$ with $v \in V_f$ are ordered in a path or a cycle by Theorem 4.4.2, it suffices to observe that $B(v, f(v))$ induces a connected subgraph in $G$ for each $v \in V_f$. ■

**Corollary 4.4.4** *For any graph $G = (V, E)$, there is an efficient optimal broadcast domination $f$ on $G$ such that a vertex $x \in V_f$ satisfies the following: $G' = G(V \setminus B(x, f(x)))$ either is empty, or is connected and has the property $\gamma_b(G') = \gamma_{bp}(G')$.*

**Proof.** By Theorem 4.4.2, let $f$ be an efficient optimal broadcast domination of $G$ such that $G_f$ is a path or a cycle. Let $x$ be any vertex of $G_f$ if $G_f$ is a cycle, any of the two endpoints of $G_f$ if $G_f$ is a path with at least two vertices, or $G_f$ itself if $G_f$ is a single vertex. Let $f'(v) = f(v)$ for all $v \in V \setminus \{x\}$. Since $f$ is efficient on $G$, $f'$ is an efficient dominating broadcast on $G'$, and $G'_{f'}$ is the result of removing $x$ from $G_f$. Thus $G'_{f'}$ is a path or empty. If $G'_{f'}$ is nonempty then $f'$ is an optimal broadcast domination on $G'$ by Lemma 4.2.3 and the corollary follows. ■

The reader should observe that while Corollaries 4.4.3 and 4.4.4 are on the form "For any graph $G$ there is an efficient optimal broadcast domination satisfying property $\Pi$" they are in fact just observing the properties of optimal efficient broadcast dominations whose domination graph is either a path or a cycle.

# Chapter 5

# The algorithm for finding an optimal broadcast domination

As we know that there is an optimal efficient broadcast domination with a domination graph that is either a path or a cycle, it is natural to split the task in two - first to find the broadcast domination with minimum cost over all efficient broadcast dominations with $G_f$ a path, then find the broadcast domination with minimum cost over all efficient broadcast dominations with $G_f$ a cycle and return the one of the two with the smallest cost. However, another division of the task into two pieces has proven to give a simpler and faster algorithm for the problem.

We divide the task into two cases as follows; either we know that $\gamma_b(G) = \gamma_{bp}(G)$ or we do not. In the first case we know there is an optimal efficient broadcast domination with $G_f$ a path. In the second there is an optimal broadcast domination with $G_f$ a either a path or a cycle. The first case seems simpler, because a path can be ordered from left to right. Such orderings are often useful when designing algorithms as they usually allow simple recurrence relations that can be used to develop Dynamic Programming or Divide and Conquer solutions. Even though our algorithm will not use any of these techniques it exploits the left to right ordering in a similar fashion. Our approach will be as follows; We find an algorithm to solve the path case, and devise a way to reduce the second case to the first.

## 5.1 Optimal Broadcast Domination of $G$ when $\gamma_b(G) = \gamma_{bp}(G)$

We are going to present an algorithm that can find a broadcast domination of minimum cost out of all efficient broadcast dominations that have a path as their domination graph. In order to find this broadcast domination we build an auxiliary graph $\mathcal{G}_u$ for every vertex $u \in V$. $\mathcal{G}_u$ is a directed graph with the property that specific directed paths in $\mathcal{G}_u$ correspond to the efficient broadcast dominations of $G$ with $G_f$ a path and $u$ contained in the ball corresponding to one of the endpoints of this path. Out of these broadcast dominations, we find the one with minimum cost by finding a minimum weight path in $\mathcal{G}_u$. Finally, we repeat this for all $u$ and pick the best $f$ ever computed.

Given a vertex $u \in V$, we define a directed graph $\mathcal{G}_u$ with weights assigned to its vertices as follows: For each $v \in V$ and each $p \in \{1, ..., rad(G)\}$, there is a vertex $(v, p)$ in $\mathcal{G}_u$ if and only if one of the following is true:

- $G(V \setminus B(v, p))$ is connected or empty, and $u \in B(v, p)$
- $G(V \setminus B(v, p))$ has at most two connected components, and $u \notin B(v, p)$.

Thus $\mathcal{G}_u$ has a total of at most $n \cdot rad(G)$ vertices. Following Corollaries 4.4.3 and 4.4.4, each vertex $(v, p)$ represents the situation that $f(v) = p$ in the broadcast domination $f$ that we are aiming to compute. We define the *weight* of each vertex $(v, p)$ to be $p$.

The role of $u$ is to define the "left" endpoint of the path that we will compute. All edges will be directed from "left" to "right". We partition the vertex set of $\mathcal{G}_u$ into four subsets:

- $A_u = \{(v, p) \mid G(V \setminus B(v, p))$ is connected and $u \in B(v, p)\}$
- $B_u = \{(v, p) \mid G(V \setminus B(v, p))$ has two connected components$\}$
- $C_u = \{(v, p) \mid G(V \setminus B(v, p))$ is connected and $u \notin B(v, p)\}$
- $D_u = \{(v, p) \mid B(v, p) = V\}$

For each vertex $(v, p)$, let $L_u(v, p)$ be the connected component of $G(V \setminus B(v, p))$ that contains $u$ (i.e., the component to the ""left" of $B(v, p)$), and left $R_u(v, p)$ be the connected component of $G(V \setminus B(v, p))$ that does not contain $u$ (i.e., the component to the "right" of $B(v, p)$). Thus $L_u(v, p) = \emptyset$ for every $(v, p) \in A_u \cup D_u$, and $R_u(v, p) = \emptyset$ for every $(v, p) \in C_u \cup D_u$.

The edges of $\mathcal{G}_u$ are directed and defined as follows: A directed edge $(v, p) \to (w, q)$ is an edge of $\mathcal{G}_u$ if and only if all of the following three conditions are satisfied:

- $B(v, p) \cap B(w, q) = \emptyset$ in $G$
- $R_u(v, p) \neq \emptyset$ and $L_u(w, q) \neq \emptyset$
- $(N_G(B(w, q)) \cap L_u(w, q)) \subseteq B(v, p)$ and $(N_G(B(v, p)) \cap R_u(v, p)) \subseteq B(w, q)$ in $G$.

To restate the last requirement in plain text: $B(v, p)$ must contain all neighbors of $B(w, q)$ in $L_u(w, q)$, and $B(w, q)$ must contain all neighbors of $B(v, p)$ in $R_u(v, p)$.

By the way we have defined the edges of $\mathcal{G}_u$, all vertices belonging to $A_u$ have indegree 0 and all vertices belonging to $C_u$ have outdegree 0. Hence, any path in $\mathcal{G}_u$ can contain at most one vertex from $A_u$ (which must be the starting point of the path) and at most one vertex from $C_u$ (which must be the ending point of the path). The vertices of $D_u$ are isolated, and every vertex of $D_u$ defines a radial broadcast domination on its own.

Looking at Figure 5.1 and 5.2 the reader should become familiar with the workings of the graph $\mathcal{G}_u$, and confident that a directed path from a vertex in $A_u$ to a vertex in $C_u$ indeed corresponds to a broadcast domination. Also, we encourage the reader to pick an efficient broadcast domination with $G_f$ a path at random, pick $u$ as one of the transmitters corresponding to the endpoints of $G_f$ and verify that $f$ indeed corresponds to a path from $A_u$ to $C_u$ in $\mathcal{G}_u$. While this is probably enough to get an intuition of why the approach works, it takes some work to actually prove it. First, let us justify

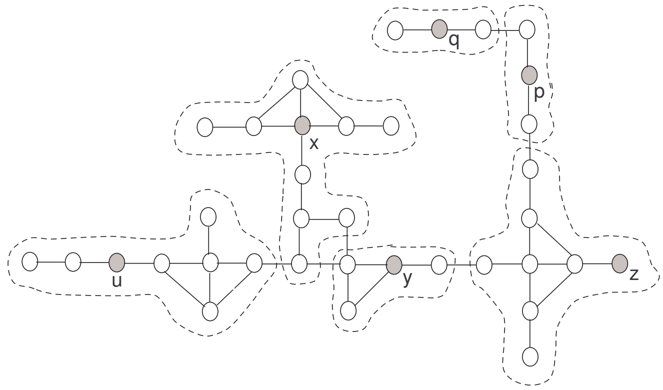


Figure 5.1: *A graph $G$ is shown. The vertex $u$ has been highlighted. So have the vertices x,y,z,p and q. $(u, 3)$ is a vertex of $\mathcal{G}_u$, in $A_u$. $(x, 3)$, $(y, 1)$, $(z, 3)$ and $(p, 1)$ are all vertices in $B_u$. $(q, 1)$ is a vertex in $C_u$. We have that $\{(u, 3), (x, 3), (y, 1), (z, 3), (p, 1), (q, 1)\}$ is a path in $\mathcal{G}_u$ from $A_u$ to $C_u$. Also, the function $f$ with $f(u) = 3$, $f(x) = 3$, $f(y) = 1$, $f(z) = 3$, $f(p) = 1$, $f(q) = 1$ and $f(v) = 0$ for all other vertices is an efficient broadcast domination with the path $\{u, x, y, z, p, q\}$ as domination graph. There are many other edges and vertices in $\mathcal{G}_u$, but we chose only show a few in order to avoid congestion on the figure.*

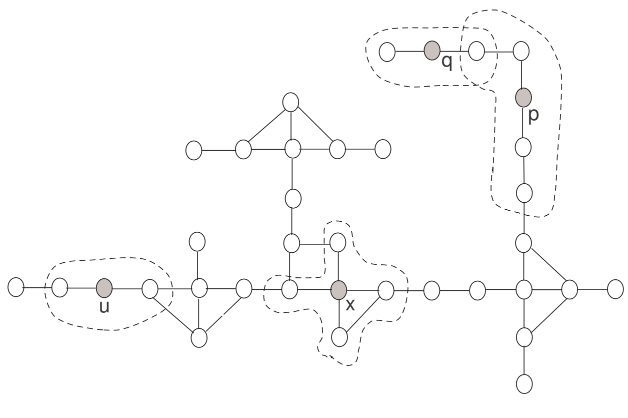


Figure 5.2: *Again, the graph $G$ is shown. The vertices u,x,p and q have been highlighted. $(u, 1)$ is* not *a vertex of $\mathcal{G}_u$ as $B(u, 2)$ contains $u$ and splits $G$ in two. $(x, 1)$ is* not *a vertex in $\mathcal{G}_u$ because $B(x, 1)$ splits $G$ into three connected components. $(p, 2)$ and $(q, 1)$ are vertices of $\mathcal{G}_u$. However there is no edge between them as they overlap.*

the intuition that edges in $\mathcal{G}_u$ go from left to right.

**Lemma 5.1.1** *Let $G$ be an arbitrary graph, and let $u$ be a vertex of $G$. Every edge in $\mathcal{G}_u$ goes from left to right, i.e., if $(v,p) \to (w,q)$ is an edge in $\mathcal{G}_u$ then $B(w,q) \subseteq R_u(v,p)$ and $B(v,p) \subseteq L_u(w,q)$.*

**Proof.** By the last two requirements in the definition of edges in $\mathcal{G}_u$, the set $B(w,q) \cap R_u(v,p)$ is nonempty. Let $x$ be a vertex in this set, and assume for contradiction that $y$ is a vertex that belongs to $B(w,q)$, but not to $R_u(v,p)$. As $B(w,q)$ induces a connected subgraph in $G$, there is a path between $x$ and $y$ in $G$ containing only vertices belonging to $B(w,q)$. On this path let $a$ be the last vertex belonging to $R_u(v,p)$, and $b$ be the next vertex on the path after $a$. As there are no edges from $R_u(v,p)$ to $L_u(v,p)$ we know that $b$ must be in $B(v,p)$, contradicting that $B(v,p) \cap B(w,q) = \emptyset$, thus proving that $B(w,q) \subseteq R_u(v,p)$. The next claim $B(v,p) \subseteq L_u(w,q)$ is proved using an identical argument. ∎

We are now ready to show that any directed path in $\mathcal{G}_u$ from $A_u \cup D_u$ to $C_u \cup D_u$ corresponds to a broadcast domination. The following observation follows directly from the definition of edges in $\mathcal{G}_u$

**Observation 5.1.2** *Given a graph $G$, and a vertex $u$ in $G$, let $P = (v_1,p_1), (v_2,p_2), \ldots, (v_k,p_k)$ be a path in $\mathcal{G}_u$ with $(v_1,p_1) \in A_u \cup D_u$ and $(v_k,p_k) \in C_u \cup D_u$. If $P$ has length at least 2, then $N_G(B(v_1,p_1)) \subseteq B(v_2,p_2)$, $N_G(B(v_k,p_k)) \subseteq B(v_{k-1},p_{k-1})$, and $N_G(B(v_i,p_i)) \subseteq B(v_{i-1},p_{i-1}) \cup B(v_{i+1},p_{i+1})$, for $1 < i < k$.*

For each path $P = (v_1,p_1),(v_2,p_2),\ldots,(v_k,p_k)$ in $\mathcal{G}_u$ with $(v_1,p_1) \in A_u \cup D_u$ and $(v_k,p_k) \in C_u \cup D_u$, let $f_P$ be the following broadcast on $G$: For every $(v_i,p_i) \in P$, $f_P(v_i) = p_i$, and $f_P(v) = 0$ for every other vertex $v$.

**Lemma 5.1.3** *Let $G$ be an arbitrary graph and let $u$ be a vertex of $G$. For every path $P$ in $\mathcal{G}_u$ from $A_u \cup D_u$ to $C_u \cup D_u$, $f_P$ is a broadcast domination on $G$.*

**Proof.** $P$ is an isolated vertex if and only if it contains a vertex from $D_u$. In this case the lemma follows trivially, as $D_u = \{(v,p) \mid B(v,p) = V\}$.

Let $P = (v_1,p_1), (v_2,p_2), \ldots, (v_k,p_k)$, and let $S = \bigcup_{j=1}^{k} B(v_j,p_j)$. We show that $S = V$. Assume for contradiction that $x \in V$ but $x \notin S$. Since $G$ is connected, there is a path from $x$ to $v_1$. Let $z$ be the first vertex on this path that is in $S$, and let $y$ be the vertex on the path before $z$. Let $j$ be such that $z \in B(v_j,p_j)$. Then $y \in N_G(B(v_j,p_j))$. By Observation 5.1.2, $y \in B(v_2,p_2)$ if $j=1$, $y \in B(v_{k-1},p_{k-1})$ if $j = k$, and $y \in B(v_{j-1},p_{j-1}) \cup B(v_{j+1},p_{j+1})$ otherwise, in any case contradicting that $z$ is the first vertex in $S$ on the path from $x$ to $v_1$. ∎

Now we want to prove that for every efficient broadcast domination $f$ on $G$, where $G_f$ is a path, there is a vertex $u$ in $G$, such that $f$ corresponds to a directed path in $\mathcal{G}_u$ that starts in $A_u \cup D_u$ and ends in $C_u \cup D_u$. More specifically, if $f$ is radial, $f$ corresponds to a single vertex in $D_u$ which we call a "path from $D_u$ to $D_u$". If $G_f$ is a path on more than one vertex the broadcast domination corresponds to a directed path from $A_u$ to $C_u$.

**Lemma 5.1.4** *Let $f$ be an efficient broadcast domination on $G$, such that $G_f = u_1, u_2, ..., u_k$ is a path. If $k = 1$ then $(u_1, f(u_1)) \in D_{u_1}$. If $k \geq 2$ then $(u_1, f(u_1)) \in A_{u_1}$, $(u_k, f(u_k)) \in C_{u_1}$, and $(u_i, f(u_i)) \in B_{u_1}$, for $1 < i < k$.*

**Proof.** Assume that $k = 1$. As $f$ is a broadcast domination, $B(u_1, f(u_1)) = V$, so $(u_1, f(u_1)) \in D_{u_1}$. Assume now that $k > 1$. Then $B(u_1, f(u_1))$ contains $u_1$. $G(V \setminus B(u_1, f(u_1)))$ is connected, as $B(u_i, f(u_i))$ is connected for every $i$, and there is an edge between every consecutive pair of balls. This gives us $(u_1, f(u_1)) \in A_{u_1}$. By the same argument $G(V \setminus B(u_k, f(u_k)))$ is connected. Also, $B(u_k, f(u_k))$ does not contain $u_1$, since $B(u_1, f(u_1))$ does, and $f$ is efficient. We conclude that $(u_k, f(u_k)) \in C_{u_1}$. For $1 < i < k$, $B(u_i, f(u_i))$ does not contain $u_1$, by the efficiency of $f$. $G(V \setminus B(u_i, f(u_i)))$ has two connected components, namely $\bigcup_{j=1}^{i-1} B(u_j, f(u_j))$ and $\bigcup_{j=i+1}^{k} B(u_j, f(u_j))$. Each of those components is connected because of the same argument as above, and there are no edges between them, as that would have yielded an edge between some $u_s$ and $u_t$ in $G_f$, with $s < i < t$, which would contradict that $G_f$ is a path. That means that $1 < i < k$ implies $(u_i, f(u_i)) \in B_{u_1}$ and the proof is complete. ∎

From the proof of Lemma 5.1.4 we can see that $L_i = \bigcup_{j=1}^{i-1} B(u_j, f(u_j))$ and $R_i = \bigcup_{j=i+1}^{k} B(u_j, f(u_j))$ are the connected components of $G(V \setminus B(u_i, f(u_i)))$, for $1 < i < k$. Now, obviously $L_i$ contains $u_1$ and $R_i$ does not, so we conclude that $L_i = L_{u_1}(u_i, f(u_i))$ and $R_i = R_{u_1}(u_i, f(u_i))$, for $1 < i < k$.

**Lemma 5.1.5** *Let $f$ be an efficient broadcast domination on $G$, such that $G_f = u_1, u_2, ..., u_k$ is a path. In $\mathcal{G}_{u_1}$ there is an edge between $(u_i, f(u_i))$ and $(u_{i+1}, f(u_{i+1}))$ for $1 \leq i < k$.*

**Proof.** $B(u_i, f(u_i)) \cap B(u_{i+1}, f(u_{i+1})) = \emptyset$ by the efficiency of $f$. $R_{u_1}(u_i, f(u_i)) = R_i$, and $R_i \neq \emptyset$, since $i < k$. $L_{u_1}(u_{i+1}, f(u_{i+1})) = L_{i+1}$, and $L_{i+1} \neq \emptyset$, since $i + 1 > 1$. For $s > i + 1$, $N_G(B(u_i, f(u_i))) \cap B(u_s, f(u_s)) = \emptyset$ because there are no edges in $G_f$ between $u_i$ and $u_s$. Thus $N_G(B(u_i, f(u_i))) \cap R_i$ is a subset of $B(u_{i+1}, f(u_{i+1}))$. The proof that $N_G(B(u_{i+1}, f(u_{i+1}))) \cap L_{i+1}$ is a subset of $B(u_i, f(u_i))$ is identical. ∎

**Corollary 5.1.6** *Let $f$ be an efficient broadcast domination on $G$, such that $G_f = u_1, u_2, ..., u_k$ is a path. Then $(u_1, f(u_1)), (u_2, f(u_2)), ...(u_k, f(u_k))$ is a directed path in $\mathcal{G}_{u_1}$, starting in $A_{u_1} \cup D_{u_1}$ and ending in $C_{u_1} \cup D_{u_1}$*

**Proof.** We have $(u_1, f(u_1)) \in A_{u_1} \cup D_{u_1}$, $(u_k, f(u_k)) \in C_{u_1} \cup D_{u_1}$, and $(u_i, f(u_i)) \in B_{u_1}$, for $1 < i < k$ by Lemma 5.1.4. By Lemma 5.1.5 there is an edge between each consecutive pair of vertices. ∎

Now the idea is to find a directed path $P_u$ in $\mathcal{G}_u$ from a vertex of $A_u \cup D_u$ to a vertex of $C_u \cup D_u$ such that the sum of the weights of the vertices of $P_u$ (including the endpoints) is minimized. Let us call this sum $W(P_u)$. Then we will compute $\mathcal{G}_u$ for each vertex $u$ in $G$, and repeat this process, and at the end choose a path with the minimum total weight. Our algorithm for the path case is as follows:

**Algorithm** Minimum Path Broadcast Domination - MPBD
**Input:** A graph $G = (V, E)$.
**Output:** An efficient broadcast domination function $f$ on $G$ with $c_f(V) \leq \gamma_{bp}(G)$.
**begin**
  **for** each vertex $v$ in $G$ **do**
    $f(v) = 0$;
  Let $P$ be a dummy path with $W(P) = rad(G) + 1$;
  **for** each vertex $u$ in $G$ **do**
    Compute $\mathcal{G}_u$ with vertex sets $A_u$, $B_u$, $C_u$, and $D_u$;
    Find a minimum weight path $P_u$ starting in a vertex of $A_u \cup D_u$ and ending in a vertex of $C_u \cup D_u$;
    **if** $W(P_u) < W(P)$ **then**
      $P = P_u$;
  **end-for**
  **for** each vertex $(v, p)$ on $P$ **do**
    $f(v) = p$;
**end**

**Theorem 5.1.7** *Given a graph $G = (V, E)$, Algorithm MPBD computes an efficient broadcast domination $f$ on $G$ such that $c_f(V) \leq \gamma_{bp}(G)$.*

**Proof.** Let $f'$ be a broadcast domination on $G$ with cost $\gamma_{bp}(G)$ and $G_{f'}$ a path. Corollary 5.1.6 assures us that $G_{f'}$ corresponds to a path $P'$ with $W(P') = \gamma_{bp}(G)$ in $\mathcal{G}_u$ for some vertex $u$ of $G$. We compute a minimum weight path $P$ in $\mathcal{G}_u$ over all $u \in V$. Thus $W(P) \leq W(P') \leq \gamma_{bp}(G)$. By Lemma 5.1.3 $P$ corresponds to a broadcast domination $f$ with $c_f(V) = W(P) \leq \gamma_{bp}(G)$. ■

**Corollary 5.1.8** *Let $G$ be a graph such that $\gamma_b(G) = \gamma_{bp}(G)$. Then Algorithm MPBD computes an efficient optimal broadcast domination on $G$.*

In a straight forward implementation of Algorithm MPBD, building $\mathcal{G}_u$ is the most time-consuming part. As $\mathcal{G}_u$ has potentially $O(n^2)$ vertices, it might have $O(n^4)$ edges. When we build $\mathcal{G}_u$ we have to check each of these potential edges for the properties of edges in $\mathcal{G}_u$. To check for the property "$B(v, p) \cap B(w, q) = \emptyset$ in $G$" we can use two breadth first searches, using $O(n + m) = O(n^2)$ time, and it is easy to see that the other properties can be checked within the same time bound. Finding a minimum weight path can be done in $O(n^4 \log n^4)$ time using a simple modification of Dijkstra's algorithm [14]. As we have to do this for every vertex $u$ of $G$, we can conclude that the running time of a straight forward implementation of Algorithm MPBD is $O(n^7)$. In the next section, we improve this running time to $O(n^4)$.

## 5.2 Improving the running time of Algorithm MPBD

In order to improve the running time, we are going to show that the number of edges of $\mathcal{G}_u$ is actually at most $n^3$, and that $\mathcal{G}_u$ is acyclic. We will use this to give an $\mathcal{O}(n^3)$ algorithm to compute a minimum weight path from $A_u \cup D_u$ to $C_u \cup D_u$. In addition, we show that the choice of $u$ does not affect $\mathcal{G}_u$ extensively, so a substantial amount of pre-computation can be done outside the outer loop over vertices $u$.

**Lemma 5.2.1** *Let $G$ be a graph on $n$ vertices, and let $u$ be any vertex of $G$. The graph $\mathcal{G}_u$ has $O(n^3)$ edges.*

**Proof.** We will need a series of claims for the proof of this lemma.

**Claim 5.2.2** *The statements $\delta_G(u,v) \leq p+q$ and $B(u,p) \cap B(v,q) \neq \emptyset$ are equivalent.*

**Proof.** Assume $x \in B(u,p)$ and $x \in B(v,q)$. Then $\delta_G(u,v) \leq \delta_G(u,x) + \delta_G(v,x) \leq p+q$. In the other direction, assume $\delta_G(u,v) \leq p+q$ and let $x$ be the vertex on a shortest path from $u$ to $v$ in $G$ having $\delta_G(u,x) = p$. As $x$ lies on a shortest path from $u$ to $v$, $\delta_G(x,v) = \delta_G(u,v) - \delta_G(u,x) \leq p+q-p = q$. Thus $x \in B(u,p)$ and $x \in B(v,q)$. ∎

**Claim 5.2.3** *Let $x$ be a vertex in $N_G(B(u,p))$. Then $x \in B(v,q)$ implies $\delta_G(u,v) \leq p+q+1$.*

**Proof.** Observe that $N_G(B(u,p)) \cap B(u,p) = B(u,p+1)$. Now $x \in B(u,p+1)$ so $B(u,p+1) \cap B(v,q)$ contains $x$ and is nonempty. By Claim 5.2.2 $\delta_G(u,v) \leq p+1+q$. ∎

**Claim 5.2.4** *If $(v,p) \to (w,q)$ is an edge in $\mathcal{G}_u$ then $\delta_G(v,w) = p+q+1$.*

**Proof.** By the requirements for an edge in $\mathcal{G}_u$, $B(v,p) \cap B(w,q) = \emptyset$ and $N_G(B(v,p)) \cap L(v,p)$ is nonempty and contained in $B(w,q)$. The first requirement implies $\delta_G(v,w) > p+q$ by Claim 5.2.2. The second implies $\delta_G(v,w) \leq p+q+1$ by Claim 5.2.3. ∎

This shows that the number of edges in $\mathcal{G}_u$ is at most $n^3$, because given $v$, $w$, and $p$, we know that $q$ must be $\delta_G(v,w) - p - 1$ if there is to be a possibility for an edge between $(v,p)$ and $(w,q)$ in $\mathcal{G}_u$. Thus the proof of Lemma 5.2.1 is complete. ∎

Now, we show that $\mathcal{G}_u$ is acyclic. Again, one should refer to the Figures 5.1 and 5.2. From those figures the fact that $\mathcal{G}_u$ is acyclic should be apparent.

**Lemma 5.2.5** *Let $G$ be a graph, and let $u$ be any vertex of $G$. The graph $\mathcal{G}_u$ is acyclic.*

**Proof.** Observe first that a cycle in $\mathcal{G}_u$ can only contain vertices from $B_u$. This is because the vertices of $D_u$ are isolated, the vertices of $A_u$ have indegree 0 and the vertices of $C_u$ have outdegree 0.

Observe next that, for an edge $(v,p) \to (w,q)$ in $\mathcal{G}_u$ with both $(v,p)$ and $(w,q)$ in $B_u$, we have $\delta_G(u,w) \geq \delta_G(u,v) - p + 1 + q$. The argument for this is as follows: $u$ is neither in $B(v,p)$ nor in $B(w,q)$. Every path from $u$ to $w$ must pass through $B(v,p)$, so it must also pass through some edge $x \to y$ with $x$ in $B(v,p)$ and $y$ in $B(w,q)$. Now $\delta_G(u,w) = \delta_G(u,x) + 1 + \delta_G(y,w)$, but $\delta_G(u,x) \geq \delta_G(u,v) - \delta_G(x,v)$ by the triangle inequality, while $\delta_G(x,v) = p$ and $\delta_G(y,w) = q$ by the second requirement for edges in $\mathcal{G}_u$.

Now, assume for the sake of contradiction that we have a cycle $(c_1,p_1),(c_2,p_2), \ldots,(c_k,p_k),(c_1,p_1)$ in $\mathcal{G}_u$. Then, by the above argument, $\delta_G(u,c_1) \geq \delta_G(u,c_k) - p_k +$

$1 + p_1 \geq \delta_G(u, c_{k-1}) - p_{k-1} + 1 + p_k - p_k + 1 + p_1 > \delta_G(u, c_{k-1}) - p_{k-1} + 1 + p_1 > ...$ $> \delta_G(u, c_1) - p_1 + 1 + p_1 = \delta_G(u, c_1) + 1$, which is an obvious contradiction. ∎

We can exploit the fact that $\mathcal{G}_u$ is acyclic in order to use a minimum weight path algorithm that is faster than Dijkstra's algorithm. We consider the minimum weight path algorithm in acyclic graphs to be a part of the folklore, but we still present a version of the algorithm, specially adapted for this setting. As $\mathcal{G}_u$ is acyclic we can find a topological ordering of $\mathcal{G}_u$ in time $\mathcal{O}(n^3)$ [14]. Given the topological ordering of the vertices we now can find a minimum weight path from $A_u \cup D_u$ to $C_u \cup D_u$ using Dynamic Programming. That is, for a vertex $(v, p)$ of $\mathcal{G}_u$ we let $\tau(v, p)$ be the weight of the minimum weight path starting in $A_u$ or $D_u$ and ending in $(v, p)$. Thus, for every vertex in $A_u$ or $D_u$, $\tau(v, p) = p$. For every other vertex, $\tau(v, p) = \infty$ if $(v, p)$ has no incoming vertices and $\tau(v, p) = p + \min q$, where the minimum is taken over all vertices $(w, q)$ so that $(w, q) \to (v, p)$ is an edge in $\mathcal{G}_u$. The weight of the minimum weight path from will $A_u \cup D_u$ to $C_u \cup D_u$ then be $\min_{(v,p) \in C_u \cup D_u} \tau(v, p)$. The minimum weight path itself can be retrieved from $\tau$ using a technique called *backtracking*. Call an edge $(v, p) \to (w, q)$ satisfying $\tau(v, p) + q = \tau(w, q)$ an *equality edge*. We can now observe that if $(v, p) \to (w, q)$ is an equality edge and $\tau(w, q) \neq \infty$ then there must be a minimum weight path from $A_u \cup D_u$ to $(w, q)$ passing through $(v, p)$.

**Algorithm** Find a minimum weight path in $\mathcal{G}_u$
**Input:** A graph $\mathcal{G}_u$, The vertex sets $A_u$, $C_u$ and $D_u$.
**Output:** A minimum weight path $P$ from $A_u \cup D_u$ to $C_u \cup D_u$.
**begin**
  Find a topological ordering of $\mathcal{G}_u$;
  **for** each vertex $(v, p)$ in $V(\mathcal{G}_u)$ **do**
    $\tau(v, p) := \infty$;
  **for** each vertex $(v, p)$ in $A_u \cup D_u$ **do**
    $\tau(v, p) := p$;
  **for** each vertex $(v, p)$ in $V(\mathcal{G}_u)$ in topological order **do**
    **for** each edge $(w, q) \to (v, p)$ in $E(\mathcal{G}_u)$ **do**
      **if** $\tau(w, q) + p < tau(v, p)$ **then**
        $\tau(v, p) := \tau(w, q) + p$;
    **end-for**
  **end-for**
  Find a vertex $(v, p)$ in $C_u \cup D_u$ minimizing $\tau$
  $(Q_v, Q_p) := (v, p)$;
  $P := \{(Q_v, Q_p)\}$;
  **while** $(Q_v, Q_p) \notin A_u \cup D_u$ **do**
    **for** each edge $(w, q) \to (Q_v, Q_p)$ in $E(\mathcal{G}_u)$ **do**
      **if** $\tau(w, q) + Q_p = \tau(Q_v, Q_p)$ **then**
        $(Q_v, Q_p) := (w, q)$;
        $P := \{(Q_v, Q_p), P\}$;
    **end-for**
  **end-do**
  Return $P$
**end**

Now, during the computation of $\tau$ each edge is considered exactly once. The same holds when we retrieve $P$. If we assume that we can check for membership in $A_u$,

$C_u$ and $D_u$ in constant time, an assumption we will justify later in this section, the algorithm above computes in $\mathcal{O}(n^3)$ steps. We have now found a way to find a minimum weight path in $\mathcal{G}_u$ in $\mathcal{O}(n^3)$ steps, an improvement from the $O(n^4 \log n^4)$ time bound in the original algorithm. However, to improve the asymptotical running time of the whole algorithm we need to build $\mathcal{G}_u$ faster.

In order to move workload outside the outer loop, we can observe that the sets $A_u$, $B_u$, $C_u$ and $D_u$ do not change so much when we change $u$. First we can notice that the set $D_u$ does not depend on the choice of $u$ at all; thus we rename this set as $D$, independent of $u$. Now we define the following sets that are independent of $u$:

- $A = \{(v,p) \mid G(V \setminus B(v,p)) \text{ is connected}\}$
- $B = \{(v,p) \mid G(V \setminus B(v,p)) \text{ has two connected components}\}$.

These sets can be computed by the following algorithm:

**Algorithm** Compute $A$, $B$, and $D$
**Input:** A graph $G = (V, E)$.
**Output:** The sets $A$, $B$, and $D$.
**begin**
    $A = \emptyset$; $B = \emptyset$; $D = \emptyset$;
    **for** each vertex $u$ in $V(G)$ **do**
        **for** each integer $k$ in $\{1, ..., rad(G)\}$ **do**
            $V' = \emptyset$;
            **for** each vertex $v$ in $V$ **do**
                **if** $\delta_G(u,v) > k$ **then**
                    $V' = V' \cup \{v\}$;
            Compute $c$, the number of connected components in $G(V')$;
            **if** $V' = \emptyset$ **then**
                $D = D \cup \{(u,k)\}$;
            **else if** $c = 1$ **then**
                $A = A \cup \{(u,k)\}$;
            **else if** $c = 2$ **then**
                $B = B \cup \{(u,k)\}$;
        **end-for**
**end**

Now, obviously $A_u \cup C_u = A$ and $B_u \subseteq B$ for any $u$. This means that, if the sets $A$ and $B$ are pre-computed, and we want to determine $A_u$ and $C_u$, we only need to iterate through every element $(v,p)$ in $A$ and determine whether to put it in $A_u$ or in $C_u$. That is done by checking whether $u \in B(v,p)$, something that can be done in constant time by checking whether $\delta_G(u,v) \leq p$, assuming that the distance between every pair of vertices in $G$ is pre-computed. To compute $B_u$ we iterate through $(v,p) \in B$, and assert that $\delta_G(u,v) > p$ in order for $(v,p)$ to be in $B_u$.

Now that we have improved the running time for finding the vertices of $\mathcal{G}_u$, we wish to accelerate the computation of the edges. In order to achieve that we need to be able to check for membership in $A_u$, $B_u$, and $C_u$ in constant time. To do this we construct an array $Member$, and the following algorithm explains this construction.

**Algorithm** Compute $A_u$, $B_u$ and $C_u$, and $Member$
**Input:** A graph $G = (V, E)$, a vertex $u$ and the sets $A$ and $B$.
**Output:** The sets $A_u$, $B_u$, $C_u$, and $Member$.
**begin**
    $A_u = \emptyset$; $B_u = \emptyset$; $C_u = \emptyset$;
    **for** every $v \in V$ and every $k \in \{1, ..., rad(G)\}$ **do**
        $Member(v, k) = \emptyset$;
    **for** each element $(v, k)$ in $A$ **do**
        **if** $\delta_G(u, v) \leq k$ **then**
            $A_u = A_u \cup \{(v, k)\}$;
            $Member(v, k) = A_u$;
        **else**
            $C_u = C_u \cup \{(v, k)\}$;
            $Member(v, k) = C_u$;
        **end-if**
    **for** each element $(v, k)$ in $B$ **do**
        **if** $\delta_G(u, v) > k$ **then**
            $B_u = B_u \cup \{(v, k)\}$;
            $Member(v, k) = B_u$;
        **end-if**
**end**

After computing the vertices of $\mathcal{G}_u$ we want to compute the edges. Recall the three conditions given on page 7 for $(v, p) \rightarrow (w, q)$ to be an edge in $\mathcal{G}_u$. We can find the edges of $\mathcal{G}_u$ by trying all possible combinations of $v$, $p$, and $w$, letting $q = \delta_G(v, w) - p - 1$ (see Claim 5.2.4), and checking the conditions for an edge from $(v, p)$ to $(w, q)$. But before we do that, we need to verify that $(v, p)$ and $(w, q)$ indeed are vertices of $\mathcal{G}_u$. This is easily done using the $Member$ array.

The first condition of being an edge is fulfilled by our choice of $q$, as $\delta_G(v, w) = p + q + 1 > p + q$. The second condition can be tested using the following observation

**Observation 5.2.6** *Let $(v, p)$ be a vertex of $\mathcal{G}_u$. Then $R_u(v, p) \neq \emptyset \Leftrightarrow (v, p) \in A_u \cup B_u$, and $L_u(v, p) \neq \emptyset \Leftrightarrow (v, p) \in B_u \cup C_u$.*

Since we can use the $Member$ array to test membership in $A_u$, $B_u$ and $C_u$ in constant time, we can use Observation 5.2.6 to test for the second condition in constant time. Now, notice that we check whether $(v, p)$ and $(w, q)$ are vertices of $\mathcal{G}_u$ twice, first before testing for the first condition, and then when testing the second. This is clearly unnecessary, so the first of these tests can be omitted.

Now we want to speed up the testing of the third condition of an edge. In order to do that, we will often need to know whether a pair of vertices lie in the same connected component in a given graph $G'$ on the form $G' = G(V \setminus B(u, p))$ where $(u, p)$ is a vertex of $\mathcal{G}_u$. By the definition of vertices in $\mathcal{G}_u$ such a graph has at most two connected components. Therefore we pre-compute a three dimensional array *Component* that assigns a 1 or a 0 to each vertex of $G(V \setminus B(u, p))$. Thus, for a vertex $(u, p)$ in $V(\mathcal{G}_u)$, $Component(u, p, v) = Component(u, p, w)$ if and only if $v$ and $w$ lie in the same connected component of $G(V \setminus B(u, p))$. This array can trivially be computed in $O(n^4)$ time by performing a depth first search in $G(V \setminus B(u, p))$ for every vertex $(u, p)$ in $\mathcal{G}_u$.

**Algorithm** Compute *Component*
**Input:** A graph $G = (V, E)$ and the set $B$.
**Output:** The array *Component*.
**begin**
  **for** every pair of vertices $u, v \in V$ and every $p \in \{1, ..., rad(G)\}$ **do**
    $Component(u, p, v) = 0$;
  **for** every $(u, p)$ in $B$ **do**
    $V' = \emptyset$;
    **for** each vertex $v$ in $V$ **do**
      **if** $\delta_G(u, v) > p$ **then**
        $V' = V' \cup \{v\}$;
    **if** $V' \neq \emptyset$ **then**
      $w$ = first element of $V'$;
      Compute $C$, the connected component of $G(V')$ containing $w$;
      **for** each vertex $v$ in $V(C)$ **do**
        $Component(u, p, v) = 1$;
    **end-if**
  **end-for**
**end**

In the following discussion, we will say that a pair of vertices of $\mathcal{G}_u$ form an *edge candidate* if $G(V \setminus B(v,p))$ has a nonempty component $C_v$ and $G(V \setminus B(w,q))$ has a nonempty component $C_w$ such that $(N_G(B(w,q)) \cap C_w)) \subseteq B(v,p)$ and $(N_G(B(v,p)) \cap C_v) \subseteq B(w,q)$ in $G$.

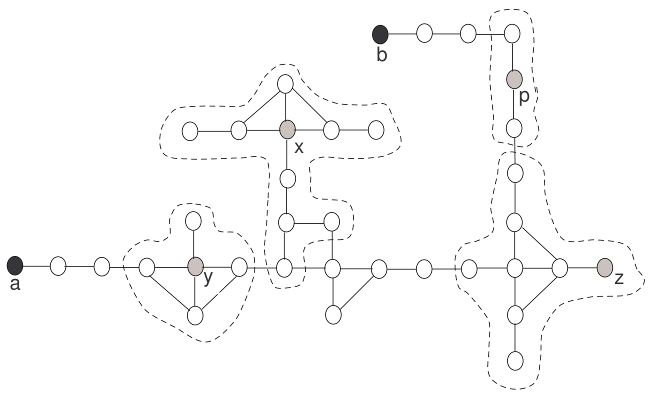


Figure 5.3: *The input graph $G$ is depicted. The vertices x,y,p and z have been highlighted. Also the endpoint vertices a and b have been colored black. Now, $(x,3)$, $(y,1)$, $(z,3)$ and $(p,1)$ are all in the set $B$. $((y,1),(x,3))$ and $((z,3),(p,1))$ form edge candidates. Notice that if $u = a$ then $(y,1) \to (x,3))$ and $(z,3) \to (p,1)$ are edges in $\mathcal{G}_u$. If $u = b$ then the edges go the other way. If $u = y$ then $(z,3) \to (p,1)$ is an edge in $\mathcal{G}_u$ while $(y,1) \to (x,3))$ is not.*

**Observation 5.2.7** *If $(v,p) \to (w,q)$ is an edge in $\mathcal{G}_u$ then the pair $(v,p),(w,q)$ is an edge candidate.*

**Proof.** We let $C_v = R_u(v, p)$ and $C_w = L_u(w, q)$, and the proof follows. ■

Now, we see that one can test whether a pair $(v, p)$, $(w, q)$ is an edge candidate outside the loop over all $u$'s. This means that we can build a three dimensional boolean array *Candidate* where an entry $Candidate(v, w, p)$ is set to true if and only if the pair $(v, p)$ and $(w, \delta_G(v, w) - p - 1)$ form an edge candidate. This array can be built in $O(n^4)$ time, as we can let $q = \delta_G(v, w) - p - 1$, scan all the neighbors of $B(v, p)$ in $G(V \setminus B(v, p))$ in the component labelled 0, and check whether they are contained in $B(w, q)$. We repeat the scan in the component labelled 1. Now we are done checking whether there exists a $C_v$ so that $(N_G(B(v, p)) \cap C_v)) \subseteq B(w, q)$. The other part of the requirement can be checked similarly.

**Algorithm** Compute *Candidate*
**Input:** A graph $G = (V, E)$ and the array *Component*.
**Output:** The array *Candidate*.
**begin**
    **for** every pair $v, w \in V$ and every $p \in \{1...rad(G)\}$ **do**
        $q = \delta_G(v, w) - p - 1$;
        $isCandidateV0 =$ **true**;
        $isCandidateV1 =$ **true**;
        $isCandidateW0 =$ **true**;
        $isCandidateW1 =$ **true**;
        **for** every $x$ in $V$ **do**
            **if** $\delta_G(v, x) = p + 1 \wedge Component(v, p, x) = 0 \wedge \delta_G(w, x) > q$ **then**
                $isCandidateV0 =$ **false**;
            **if** $\delta_G(v, x) = p + 1 \wedge Component(v, p, x) = 1 \wedge \delta_G(w, x) > q$ **then**
                $isCandidateV1 =$ **false**;
            **if** $\delta_G(w, x) = q + 1 \wedge Component(w, q, x) = 0 \wedge \delta_G(v, x) > p$ **then**
                $isCandidateW0 =$ **false**;
            **if** $\delta_G(w, x) = q + 1 \wedge Component(w, q, x) = 1 \wedge \delta_G(v, x) > p$ **then**
                $isCandidateW1 =$ **false**;
        **end-for**
        $Candidate(v, w, p) = (isCandidateV0 \vee isCandidateV1) \wedge (isCandidateW0 \vee isCandidateW1)$;
    **end-for**
**end**

The next lemma shows how this structure can be used.

**Lemma 5.2.8** *There is an edge $(v, p) \to (w, q)$ in $\mathcal{G}_u$ if and only if $(v, p)$ and $(w, q)$ are contained in $A_u \cup B_u \cup C_u$, fulfill the first two conditions for an edge in $\mathcal{G}_u$, form an edge candidate, and satisfy $w \in R_u(v, p) \wedge v \in L_u(w, q)$.*

**Proof.** If there is an edge from $(v, p)$ to $(w, q)$ then one can easily confirm that all requirements in the lemma are met. In the other direction, we see that the first two conditions for an edge are fulfilled. As for the third, we have that they form an edge candidate, so it holds to show that $C_v$ is in fact $R_u(v, p)$ while $C_w$ is $L_u(w, q)$. Assume for contradiction that $C_v$ is $L_u(v, p)$. Then $B(w, q)$ contains vertices both in $L_u(v, p)$ and $R_u(v, p)$ which is impossible as $B(w, q)$ is connected while $L_u(v, p)$ and $R_u(v, p)$ are separate components. This means that $C_v$ indeed must be $R_u(v, p)$. The proof that $C_w$ is $L_u(w, q)$ uses the same strategy and the lemma follows. ■

In order to check for an edge in $\mathcal{G}_u$ one can use the requirements in Lemma 5.2.8. If we make use of our pre-computed arrays we can check whether a given pair $(v,p)$, $(w,q)$ fits these requirements in constant time. We are now ready to give the details of a more efficient version of Algorithm MPBD.

**Algorithm** Revised Minimum Path Broadcast Domination - RMPBD
**Input:** A graph $G = (V, E)$.
**Output:** An efficient broadcast domination function $f$ with $C_f(V) \leq \gamma_{bp}(G)$.
**begin**
    Find the distance matrix and the radius of $G$;
    Compute $A$, $B$, and $D$;
    Compute $Component$;
    Compute $Candidate$;
    **for** each vertex $v$ in $G$ **do**
        $f(v) = 0$;
    Let $P$ be a dummy path with $W(P) = rad(G) + 1$;
    **for** each vertex $u$ in $G$ **do**
        Compute $A_u$, $B_u$, $C_u$, and $Member$;
        $E(\mathcal{G}_u) = \emptyset$;
        **for** each pair $v, w \in V$ and each $p \in \{1, ..., \delta_G(v,w) - 3\}$ **do**
            $q = \delta_G(v,w) - p - 1$;
            **if not** $(Member(v,p) = A_u \vee Member(v,p) = B_u)$
                $\wedge(Member(w,q) = B_u \vee Member(w,q) = C_u)$ **then continue**
            **if not** $Candidate(v,w,p)$ **then continue**
            **if** $\delta_G(u,v) - p \leq \delta_G(u,w) - q$ **then**
                $E(\mathcal{G}_u) = E(\mathcal{G}_u) \cup ((v,p) \to (w,q))$;
            **else**
                $E(\mathcal{G}_u) = E(\mathcal{G}_u) \cup ((w,q) \to (v,p))$;
        **end-for**
        Compute a minimum weight path $P_u$ starting in a vertex of $A_u \cup D$ and ending in a vertex of $C_u \cup D$;
        **if** $W(P_u) < W(P)$ **then**
            $P = P_u$;
    **end-for**
    **for** each vertex $(v,p)$ on $P$ **do**
        $f(v) = p$;
**end**

The correctness of this algorithm follows directly from Lemma 5.1.7, because this algorithm does exactly the same things as the one proposed in the previous section. The complexity analysis is deduced from the discussion above.

**Lemma 5.2.9** *The running time of Algorithm RMPBD on a graph $G$ with $n$ vertices is $O(n^4)$.*

**Proof.** From our discussion above, it should be clear that all the pre-computation can be done in $O(n^4)$ time. In the loop over vertices $u$ we can compute the sets $A_u$, $B_u$ and $C_u$ using $O(n^2)$ operations. The edges of $\mathcal{G}_u$ are found in $O(n^3)$ time, and a minimum weight path is found in $O(n^3)$ time as well. As these operations are inside the loop over $u$ they are repeated $O(n)$ times, yielding a time complexity of $O(n^4)$ for the whole algorithm. ∎

## 5.3 Optimal Broadcast Domination for all cases

Now we want to compute an optimal broadcast domination for any given graph $G$. Our approach will be as follows. Let $x$ be any vertex of $G$. For each $k$ between 1 and $rad(G)$ such that $G' = G(V \setminus B(x,k))$ is connected or empty, we run the Minimum Path Broadcast Domination algorithm RMPBD on $G'$. Our algorithm for the general case is given below.

**Algorithm** Optimal Broadcast Domination - OBD
**Input:** A graph $G = (V, E)$.
**Output:** An optimal broadcast domination function $f$ on $G$.
**begin**
    $opt = rad(G) + 1$;
    **for** each vertex $x$ in $G$ **do**
        **for** $k = 1$ **to** $rad(G)$ **do**
            **if** $G' = G(V \setminus B(x,k))$ is connected or empty **then**
                $f = \text{RMPBD}(G')$;
                **if** $c_f(V \setminus B(x,k)) + k < opt$ **then**
                    $opt = c_f(V \setminus B(x,k)) + k$;
                    $f(x) = k$;
                    **for** each vertex $v$ in $B(x,k) \setminus \{x\}$ **do**
                        $f(v) = 0$;
                **end-if**
            **end-if**
**end**

In this way, we consider all broadcast dominations $f$ whose corresponding domination graphs are paths or cycles. The advantage of this approach is its simplicity. The disadvantage is that we also consider many cases that do not correspond to a path or a cycle, which we could have detected with a longer and more involved algorithm. However, these unnecessary cases do not threaten the correctness of the algorithm, and detecting them does not decrease the asymptotic time bound.

**Theorem 5.3.1** *Algorithm OBD computes an optimal broadcast domination of any given graph.*

**Proof.** Let $G = (V, E)$ be the input graph. By Theorem 4.4.2 and Corollary 4.4.4, there is a vertex $x$ in $V$ and an integer $k \in [1, rad(G)]$ such that the graph $G' = G(V \setminus B(x,k))$ has an efficient optimal broadcast domination $f'$ where the domination graph $G'_{f'}$ is a path, and that $f'$ can be extended to an optimal broadcast domination $f$ for $G$ with $f(x) = k$, $f(v) = 0$ for $v \in B(x,k)$ with $x \neq v$, and $f(v) = f'(v)$ for all other vertices $v$. Algorithm RMPBD computes an optimal broadcast domination of $G'$, and since Algorithm OBD tries all possibilities for $(x,k)$, the result follows. ∎

Note that although there is always an efficient optimal broadcast domination $f$ such that $G_f$ is a cycle or a path, there can of course exist other optimal broadcast dominations $f'$ with $c_{f'}(V) = c_f(V)$ such that $G_{f'}$ is not a path or a cycle, and such that $f'$ is not efficient. The optimal broadcast domination returned by algorithm OBD does not necessarily correspond to a path or a cycle, since we do not force the endpoints (or forbid the interior points) of the path for $G'$ to be neighbors of $B(x, k)$. Nor is the returned broadcast necessarily efficient, as some ball $B(v, p)$ might have an outreach outside of $G'$ and might overlap with $B(x, k)$.

**Theorem 5.3.2** *The running time of Algorithm OBD on a graph $G$ with $n$ vertices is $O(n^6)$.*

**Proof.** First, algorithm OBD finds the radius of $G$, which can be done in $O(n^3)$ time. For every iteration of the inner loop we find out whether $G'$ is connected, and call algorithm RMPBD. The first of these tasks can be done in $O(n + m) = O(n^2)$ time, while the second is done in $O(n^4)$ time by Lemma 5.2.9. The inner loop iterates $O(n^2)$ times, and the proof is complete. ∎

The polynomial time algorithm to solve Optimal Broadcast Domination is the main result of this thesis. The work, without the part on optimizing up algorithm MPBD, was presented at the 31st International Workshop on Graph-Theoretic Concepts in Computer Science [12]. We will now proceed to consider a couple variations of the problem.

# Chapter 6

# Generalizations of Broadcast Domination

Having coped with the original form of Broadcast Domination we wish to study more general and realistic versions. For instance, our original motivation to study Broadcast Domination was to devise a model that we could use to optimize placement of transmitters or radio stations. In our model, building and operating a transmitter that can send a signal $p$ units away from itself has a cost of $p$. This is an assumption that is natural because it yields the simplest model. However, this assumption does not hold from a physical point of view. Since a transmitter sends a signal away from itself in all directions, the energy of the transmitted signal is inverse proportional to the square of the distance to the transmitter. Thus, sending a signal $p$ distance units would require $\mathcal{O}(p^2)$ units of energy - and $\mathcal{O}(p^2)$ units of money to pay for this energy.

## 6.1 Broadcast Domination with other cost functions

A way to extend our model in order to handle this problem, is to provide a cost function $\zeta : Z^+ \rightarrow Z^+$. The interpretation of $\zeta$ will be that a dominator of power $p$ costs $\zeta(p)$. That is, we redefine $C_f(S)$ to be $\sum_{v \in S} \zeta(f(v))$. As not building anything at all doesn't cost anything we will assume that $\zeta(0) = 0$. Also, we will assume that $\zeta$ is a nondecreasing function, as it seems illogical that a stronger transmitter would be cheaper than a weak. Now, we still wish to find the broadcast domination of $G$ that minimizes $C_f(V)$, the only difference is that the definition $C_f$ has changed. We will use $\gamma_{bp-\zeta}(G)$ to denote the minimum cost of a broadcast domination of $G$ when using $\zeta$ as a cost function.

If we let the cost function $\zeta$ be a part of the input, the problem immediately becomes $\mathcal{NP}$-complete. We can show this by giving a simple reduction from Dominating Set.

**Lemma 6.1.1** *Dominating Set $\leq_T$ Broadcast Domination with General Cost Functions.*

**Proof.** For a given graph $G = (V, E)$ as input to Dominating Set, use $G$ as input for Broadcast Domination with General Cost Functions. Also, let $\zeta(1) = 1$ and $\zeta(k) = |V| + 1$ for $k > 1$. Now, in any optimal broadcast domination $f$, $f(v) <= 1$ for all vertices $v$. This is because using a dominator of power 2 would cost more than making all vertices dominators with power 1. But Broadcast Domination with power restricted to be less than or equal to 1 is just Dominating Set ∎

Since letting the cost function be part of the input makes the problem to hard to handle, we wish to concentrate on specific cost functions, and classes of cost functions. First, observe that letting $\zeta(k) = k$ returns us to the original Broadcast Domination. In this case, we obviously can apply algorithm OBD. It is interesting to see whether there are other cost functions for which this algorithm can be used. The answer to this question is yes, and actually, the algorithm can be used on quite a wide range of cost functions.

We only give an outline of the proof of this claim. A rigorous proof would require us to go through all the calculations again, but using the new definition of $C_f$ instead of the old one. This would require a large amount of space, and it should be fairly obvious that these calculations indeed do go through.

**Claim 6.1.2** *If $\zeta(a) + \zeta(b) \geq \zeta(a + b)$ for all integers $a$ and $b$, Algorithm OBD gives the correct answer for Broadcast Domination with cost function $\zeta$.*

**Proof.** The algorithm is based on the existence of an optimal efficient broadcast domination whose domination graph is either a path or a cycle. In the proof of the existence of such a broadcast domination we used the fact that if an optimal broadcast domination violates one of these conditions we could replace some of the transmitters with a strong one while keeping the total cost at the same level. In the original setting $\zeta(a) + \zeta(b) = \zeta(a + b)$, while in the new setting $\zeta(a) + \zeta(b) \geq \zeta(a + b)$. This means that replacing many weak transmitters by one strong will cost us at most as much as it did when $\zeta(k) = k$. Hence the proofs of Lemma 4.2.1 and Theorem 4.4.2 will still go through. Having confirmed that these results still hold, one can easily check that the algorithm still works ∎

This result might seem uplifting - we now have many cost functions for which we can apply the algorithm. Unfortunately, these functions all grow slower than $\zeta(k) = k$, while the cost functions that we argued to be the most realistic from a physical perspective all grow a lot faster than $\zeta(k) = k$. And the bad news just keeps coming. Not only does the algorithm give incorrect results for cost functions that grow asymptotically faster than $\zeta(k) = k$. The algorithm even breaks down when $\zeta(k) = 2k - 1$ for $k > 0$ and $\zeta(0) = 0$. The reason for this is that the efficiency result from Dunbar et al. no longer applies.

**Observation 6.1.3** *If $\zeta(k) = 2k - 1$ for $k > 0$, $\zeta(0) = 0$, the only optimal broadcast domination for the graph on Figure 6.1 is inefficient.*

As the efficiency result is so crucial to the presented algorithm for Broadcast Domination, it seems improbable that a modified version of algorithm OBD might tackle these cost functions. At the same time, that does not mean that Broadcast Domination is $\mathcal{NP}$-complete for all these cost functions. We give a result that shows that faster growing cost functions are harder to tackle than functions that grow slow.

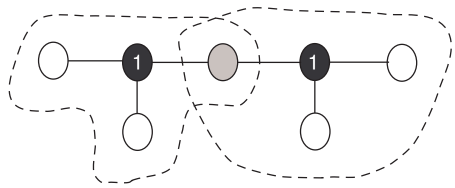


Figure 6.1: *This is clearly the only way to dominate the graph only using two dominators with power* 1. *A single dominator with power* 2 *costs more than two dominators with power* 1

The idea is as follows: In Broadcast Domination we are free to use as powerful dominators as we want. If we make the powerful dominators to costly to be used we would have to dominate the graph only using weak dominators. Then the problem of doing this at minimum cost would start to resemble Dominating Set - an $\mathcal{NP}$-complete problem. In fact, we are going to show that if the cost function grows fast enough, and the graph is sparse enough, finding an optimal broadcast domination with the given cost function is equivalent to finding a minimum cardinality dominating set in that graph.

**Lemma 6.1.4** *If* $\zeta(1) \cdot \gamma(B(v,p)) \leq \zeta(p)$ *then* $\gamma_{bp-\zeta}(G) = \gamma(G) \cdot \zeta(1)$

**Proof.** First of all, $\gamma_{bp-\zeta}(G) \leq \gamma(G) \cdot \zeta(1)$ as any dominating set corresponds to a broadcast domination with all dominators having power 1. Second, suppose you have an optimal broadcast domination $f$ with respect to $\zeta$. Then, for each $v \in V_f$ let $D_v$ be a minimum cardinality dominating set of $B(v, f(v))$. Now, $D = \bigcup_{v \in V_f} D_v$ is a dominating set of $G$. This yields the chain of inequalities $\zeta(1) \times \gamma(G) \leq \zeta(1) \times |D| \leq \zeta(1) \times \sum_{v \in V_f} |D_v| = \sum_{v \in V_f} \zeta(1) \times \gamma(|B(v, f(v))|) \leq \sum_{v \in V_f} \zeta(f(v)) = \gamma_{bp-\zeta}(G)$ ▪

If $\zeta(1) = 1$ which is often the case, one can simplify the equations in lemma 6.1.4

**Corollary 6.1.5** *If* $\zeta(1) = 1$ *and* $\gamma(B(v,p)) \leq \zeta(p)$ *then* $\gamma_{bp-\zeta}(G) = \gamma(G)$

Since determining $\gamma(G)$ is an $\mathcal{NP}$-complete problem, we can use Corollary 6.1.5 to give a class of cost functions for which Broadcast Domination is in $\mathcal{NPC}$.

**Lemma 6.1.6** *If Dominating Set is in* $\mathcal{NPC}$ *when input is restricted to a graph class with all member graphs having* $\gamma(B(v,p)) \leq \zeta(p)$, $\zeta(1) = 1$ *then Broadcast Domination with cost function* $\zeta$ *is* $\mathcal{NP}$*-complete*

**Proof.** We give a reduction from Dominating Set with restricted input to Broadcast Domination with $\zeta$ as cost function. The trick is simply to use the same graph $G$ as input to both problems. By Corollary 6.1.5 $\gamma_{bp-\zeta}(G) = \gamma(G)$. Thus by determining $\gamma_{bp-\zeta}(G)$ we automatically determine $\gamma(G)$ ▪

A simple consequence of Lemma 6.1.6 is that there is some threshold cost function $\eta$ so that Broadcast Domination is $\mathcal{NP}$-complete with all cost functions $\zeta \geq \eta$. Finally, we are going to study Quadratic Broadcast Domination, Broadcast Domination with the quadratic cost function $\zeta(p) = p^2$. We are going to use Lemma 6.1.6 to give a connection between Quadratic Dominating Set and Grid Dominating Set - Dominating Set when the input graphs are restricted to subgraphs of grids.

An *m by n grid graph* is a graph $G = (V, E)$ where $V$ is the cartesian product of $\{1, 2, ...m\}$ and $\{1, 2, ...n\}$. There is an edge $((x_1, y_1), (x_2, y_2)) \in E$ if $|x_1 - x_2| + |y_1 - y_2| = 1$ A simple way to envision an $m$ by $n$ grid graph is just an $m$ by $n$ rectangular grid with the lines representing edges and the intersection points being the vertices. When talking about vertices in a grid graph, we say that a vertex $(x, y)$ is *even* if $x + y$ is an even number, and *odd* otherwise. We define the set of all even grid vertices $V_E = \{(x, y) \in \mathbb{Z}^2 : x+y \text{ is even}\}$ and similarly the set of all odd ones, $V_O = \{(x, y) \in \mathbb{Z}^2 : x + y \text{ is odd}\}$. Observe that in a subgraph of a grid, an even vertex can only have odd neighbors, and an odd vertex can only have even.

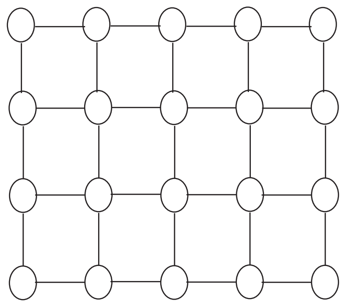

Figure 6.2: *A* 4 *by* 5 *grid.*

A ball in a grid graph has a very special structure. In fact, in a grid graph $G$, $B((x, y), p)$ is just $V \cap \{(x', y') : |x - x'| + |y - y'| \leq p\}$. If $G$ is a connected subgraph of a grid, we can see that for a given ball $B((x, y), p)$ both $B((x, y), p) \cap V_E$ and $B((x, y), p) \cap V_O$ are dominating sets of $B((x, y), p)$. This is because every vertex in a ball has a neighbor inside the ball, and every even vertex has only odd neighbors and vice versa.

**Observation 6.1.7** *Let $k$ be the minimum of $|\{(x', y') : |x - x'| + |y - y'| \leq p\} \cap V_E|$ and $|\{(x', y') : |x - x'| + |y - y'| \leq p\} \cap V_O|$. Then $k = p^2$*

**Proof.** First, observe that $k$ is independent of the choice of $x$ and $y$. Hence we only need to prove the statement for $x = y = 0$. We prove that $|\{(x', y') : |x'| + |y'| \leq p\} \cap V_E| = p^2$ if $p$ is odd by induction. For $p = 1$ this is trivially true. Assume true for a given odd $p$. Then $|\{(x', y') : |x'| + |y'| \leq p + 2\} \cap V_E| = |\{(x', y') : |x'| + |y'| \leq p\} \cap V_E| + |\{(x', y') : |x'| + |y'| = p + 2\}|$. Now, $|\{(x', y') : |x'| + |y'| = p + 2\}| = 4p + 4$. Hence $|\{(x', y') : |x'| + |y'| \leq p + 2\} \cap V_E| = p^2 + 4p + 4 = (p + 2)^2$. Using a similar proof we can show that $|\{(x', y') : |x'| + |y'| \leq p\} \cap V_O| = p^2$ if $p$ is even. Also, by similar arguments $|\{(x', y') : |x'| + |y'| \leq p\} \cap V_E| = (p + 1)^2$ when $p$ is even and $|\{(x', y') : |x'| + |y'| \leq p\} \cap V_O| = (p + 1)^2$ if $p$ is odd. Together these results complete the proof. ∎

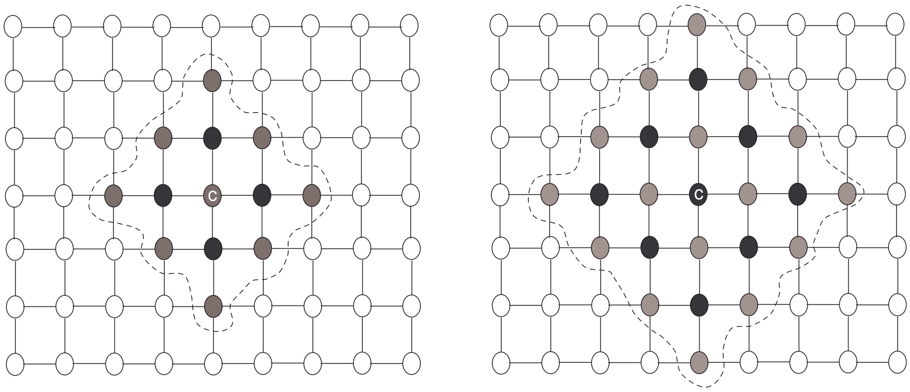


Figure 6.3: *Two balls in a grid. To the left a ball with radius* 2 *and a dominating set of size* 4*. To the right a ball with radius* 3 *with a dominating set of size* 9*. The center of the balls is marked by a C. The grey balls are the receivers, the black ones the dominators.*

**Lemma 6.1.8** *If $G$ is a subgraph of a grid, then $\gamma(B((x,y),p)) \leq p^2$.*

**Proof.** As $G$ is a subgraph of a grid, $D_e = B((x,y),p) \cap V_E$ and $D_o = B((x,y),p) \cap V_O$ are dominating sets of $B((x,y),p)$ Now, $D_e = B((x,y),p) \cap V_E \subseteq \{(x',y') : |x-x'| + |y-y'| \leq p\} \cap V_E \cap V \subseteq \{(x',y') : |x-x'| + |y-y'| \leq p\} \cap V_E$. Similarly $D_o \subseteq \{(x',y') : |x-x'| + |y-y'| \leq p\} \cap V_O$. Hence, by Observation 6.1.7, $\min(|D_e|, |D_o|) \leq p^2$. ▪

**Theorem 6.1.9** *If Grid Dominating Set is $\mathcal{NP}$-complete then Quadratic Broadcast Domination is $\mathcal{NP}$-complete as well.*

**Proof.** By Lemma 6.1.8 subgraphs of grids satisfy $\gamma(B((x,y),p)) \leq p^2$. Now, assume Grid Dominating Set is $\mathcal{NP}$-complete. Thus, by Lemma 6.1.6 Quadratic Broadcast Domination $\mathcal{NP}$-complete. ▪

To complete the proof that Quadratic Broadcast Domination indeed is $\mathcal{NP}$-complete we need to show that Grid Dominating Set is in $\mathcal{NPC}$. We do not have this result, but we can get somewhere along the road to it by providing a conjecture that if proven true will prove the $\mathcal{NP}$-completeness of Grid Dominating Set. We will call an edge $(u,v)$ *contractible* if the following three conditions are satisfied:

- $d(u) = d(v) = 2$
- $N[u] \neq N[v]$
- If $x \in N[u] \setminus N[v]$ and $y \in N[u] \setminus N[v]$ then $(u,v) \notin E$

Notice that given that the two first conditions are satisfied, the vertices $x$ and $y$ are uniquely defined. Now, if $(u,v)$ is a contractible edge we let $G_{uv} = (V \setminus \{u,v,y\}, E \setminus (\{(x,u),(u,v)\} \cup \{(y,k) : k \in V\}) \cup \{(x,k) : (y,k) \in E\})$. This corresponds to taking $G$ and contracting the vertices $x$,$u$,$v$ and $y$ into one vertex.

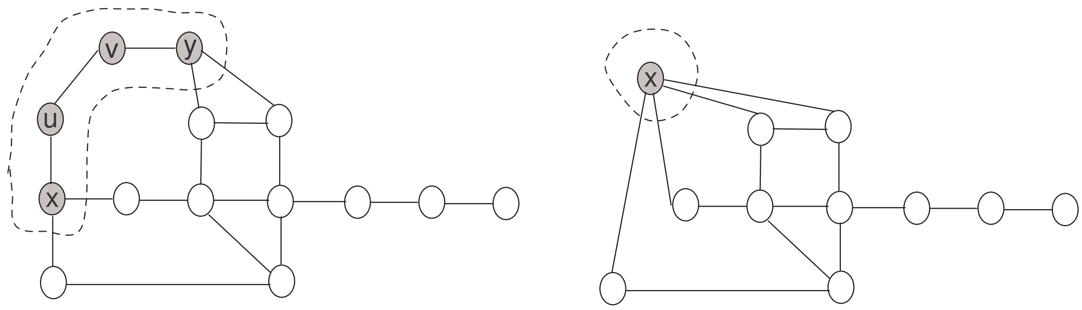


Figure 6.4: *On the left hand side we have G with the vertices x,u,v and y highlighted. On the right is $G_{uv}$ which is obtained by contracting u,v and y into x.*

**Lemma 6.1.10** *If $G$ has a contractible edge $(u, v)$ then $\gamma(G) = k + 1 \Leftrightarrow \gamma(G_{uv}) = k$.*

**Proof.** Suppose $G_{uv}$ has a dominating set $D'$ of size $k$. If $x \in D'$ then $D = D' \cup \{y\}$ is a dominating set of size $k + 1$. If $x \notin D'$ then $x$ has a neighbor $k$ in $D'$. Now, in $G$ either $(k, x) \in E$ or $(k, y) \in E$. In the first case $D = D' \cup v$, in the second $D = D' \cup u$ is a dominating set of $G$ of size $k + 1$.

In the other direction, suppose $G$ has a dominating set $D$ of size $k + 1$. If $|D \cap \{x, u, v, y\}| \geq 2$ then $D' = D \cup \{x\} \setminus \{u, v, y\}$ is a dominating set of $G_{uv}$ with size $k$. If $|D \cap \{x, u, v, y\}| = 1$ then $|D \cap \{u, v\}| = 1$ as $x$ cannot dominate $v$ and $y$ cannot dominate $u$. Also, as $u$ cannot dominate $y$ and $v$ cannot dominate $x$ we know that either $x$ or $y$ has a neighbor in $D \setminus \{x, u, v, y\}$. Hence in this case $D' = D \setminus \{x, u, v, y\}$ is a dominating set of $G_{uv}$ of size $k$. Finally observe that $D \cap \{x, u, v, y\} = \emptyset$ contradicts that $D$ is a dominating set and is impossible. ∎

We will extend our notation in order to be able to properly formulate the last results of this section. We define a relation $<_C$ on graphs as follows: $H <_C G$ if $G$ has a contractible edge $(u, v)$ so that $H = G'_{uv}$. Also, if $H <_C G$ and $H' <_C H$ then $H' <_C G$. Each time we contract, we contract 4 vertices into one. Thus, if $H < G$ it is easy to compute how many contractions are needed to go from $G$ to $H$.

**Corollary 6.1.11** *Assume $H <_C G$. Then $H$ has a dominating set of size $k$ if and only if $G$ has a dominating set of size $k + \frac{V(G)-V(H)}{3}$.*

Using this notation we can prove that Cubic Planar Dominating Set, Dominating Set when input is restricted to planar graphs with maximum degree 3, is $\mathcal{NP}$-complete. This theorem has been known for some time, however no complete proof for it has yet been published in the literature.

**Lemma 6.1.12** *For a planar graph $G$ there is a cubic planar graph $H$ satisfying $G <_C H$. $H$ can be computed from $G$ in polynomial time.*

**Proof.** Let $T(G)$ be the set of all vertices in $G$ with degree at least 4. We prove by induction over $Z(G) = \sum_{v \ T(G)} d(v)$. Obviously, if $Z(G) < 4$ we are done. Otherwise, let $x$ be a vertex of degree $\Delta(G)$. We build $G'$ as follows: Attach $x$ to the endpoint $u$ of a path $\{u, v, y\}$. Now, let $(x, w)$ and $(x, z)$ be two edges adjacent to the face that $y$ is in. Remove these edges from the graph and add the edges $(y, w)$ and $(y, z)$. Now, it is easy to see that G' is still planar. Also $(u, v)$ forms a contractible edge, and $G'_{uv} = G$. Finally, note that the degree of all vertices in $G'$ except $x$ remains the same in $G'$ as in $G$. Also, in $G'$, $d(y) = 3$, $d(u) = d(v) = 2$ and the degree of $x$ decreases by one from $G$ to $G'$. Hence $T(G') \subseteq T(G)$ and $Z(G') < Z(G)$ as the degree of $x$ decreases. As $Z(G)$ is at most $n^2$ at the start of this process, at most $n^2$ such replacements are needed to compute $H$ ∎

**Theorem 6.1.13** *Planar Dominating Set $\leq_T$ Planar Cubic Dominating Set.*

**Proof.** On input $G$ to Planar Dominating Set, use $H$ as described in Lemma 6.1.12 as input to Planar Cubic Dominating Set. By Lemma 6.1.12 $H$ is computed in polynomial time. By Corollary 6.1.11, $G$ has a dominating set of size $k$ if and only if $H$ has a dominating set of size $k + \frac{V(H)-V(G)}{3}$ ∎

We believe that one can use the way of splitting a vertex into a path on four vertices presented in Lemma 6.1.10 to make a given cubic planar graph sparse enough to be embedded on a grid. More formally, we present the following conjecture, on the same form as Lemma 6.1.12:

**Conjecture 6.1.14** *For a cubic planar graph $G$ there is a graph $H$ that subgraph of a grid, satisfying $G <_C H$. $H$ can be computed from $G$ in polynomial time.*

If the conjecture is true, it follows that Grid Dominating Set is in $\mathcal{NPC}$ which in turn implies $\mathcal{NP}$-completeness of Quadratic Broadcast Domination.

## 6.2 Weighted Broadcast Domination

A natural generalization of Broadcast Domination problem is to consider the problem on weighted graphs. We return to the original cost model, that is we again let $C_f(S)$ be defined as $\sum_{v \in S} f(v)$. The difference between Weighted Broadcast Domination and the original problem, is that the input graph $G$ now has weights $w : E \to N$ associated with the edges. We redefine the *length* of a path to be the sum of the weights of all edges in the path. The *distance* $\delta(u, v)$ between two vertices $u$ and $v$ in a weighted graph is still the length of a shortest path between $u$ and $v$.

Having redefined distance we need to readjust our notion of broadcast dominations and related objects to the new situation. In this setting, a *broadcast domination* is defined just as in the unweighted case, as a function $f : V \to \mathbb{Z}^+$ satisfying that for every $u \in V$ there is a $v \in V_f$ satisfying $\delta(u, v) \leq f(v)$. The *cost* of a broadcast domination is $C_f(V)$ and the minimum cost over all broadcast dominations will be denoted $\gamma_{bw}(G)$. A ball $B(v, p)$ with center $v$ and radius $r$ defined as in the unweighted model, $B(v, p) = \{u : \delta(u, v) \leq p\}$.

Unfortunately, the similarities to the original problem seem end at the definitions. Just as for Quadratic Broadcast Domination, it is easy to provide an example where the only optimal broadcast domination is inefficient.

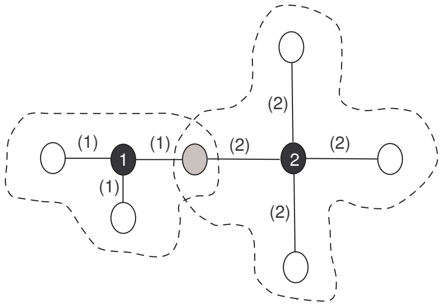


Figure 6.5: *No other broadcast domination with cost* 3 *exists. Black vertices are transmitters with their strength indicated inside them. The weight of the edges is shown in parentheses besides the edges*

Again, the radius of $G$, $rad(G)$ provides an upper bound on $\gamma_b p(G)$. However, in this case, the radius can be a bad upper bound as edges can have enormous weighs. Just consider a graph $G$ with an edge of weight $1000n$ between every pair of vertices. The optimal broadcast domination is just a function that assigns 1 to every vertex. This broadcast domination has cost $n$, while the radius of $G$ is $1000n$. Because of this example, we refine the upper bound to be $\gamma_{bw}(G) \leq \min(n, rad(G))$. One should note that as a consequence, an optimal broadcast domination will never have a vertex $v$ with $f(v) > n$. Because of this, edges with weight larger than $n$ can be safely deleted from the input graph. Finally, the example above shows that $\frac{diam(G)}{3} \leq \gamma_{bw}(G)$ no longer holds for weighted graphs.

Even though we do not have an efficiency result for the weighted case, it seems hard to devise a proof that Weighted Broadcast Domination is $\mathcal{NP}$-complete. In fact it seems that one in Weighted Broadcast Domination still gets optimal solutions that have a certain structure. We wish to formalize this intuition. In order to do this, we generalize domination graphs to weighted graphs. In a domination graph the dominators are vertices, and there is an edge between dominators if they are "close enough" in the graph. In the original setting, "close enough" meant that $(u, v)$ was an edge in $G_f$ if $\delta(u, v) = f(u) + f(v) + 1$. However we do not have efficiency when the input graphs are weighted. Hence it is reasonable to say that $(u, v)$ is an edge in $G_f$ if $\delta(u, v) \leq f(u) + f(v) + 1$.

If we attempt to go through with the proof of Theorem 4.4.2 we see that the argument in the case where $f(x) + f(y) > f(z)$ is still valid, while the one in the case that $f(z) \geq f(x) + f(y)$, is not. We can use the valid part to show the existence of an optimal solution where all vertices in the domination graph have degree $\mathcal{O}(\log n)$.

**Lemma 6.2.1** *There is an optimal broadcast domination $f$ of the weighted graph $G$ satisfying that if all neighbors of a transmitter $v$ in $G_f$, $x_1 \ldots . x_k$, are sorted so that $f(x_i) \leq f(x_{i+1})$, then $f(x_j) \geq \sum_{i=1}^{j-1} f(x_i)$.*

**Proof.** We show that if $f$ violates the condition above, we can build a new broadcast domination $f'$ with $C_{f'}(V) = C_f(V)$ and $|V'_f| < |V_f|$. The result then follows by induction.

Suppose $f$ violates the condition in the lemma. Then there must be a vertex $v \in V_f$ satisfying the following: If $v$'s neighbors $x_1....x_k$ in $G_f$ are sorted by power in nondecreasing order, there is an integer $p'$ so that $f(x_{p'}) < \sum_{i=1}^{p'-1} f(x_i)$. In this case, let $p$ be the smallest possible integer so that $f(x_p) < \sum_{i=1}^{p-1} f(x_i)$. Observe that $p \geq 3$, as $0 \leq x_1 \leq x_2$. Now, we let $f'(u) = f(u)$ for all $u \in V\ (v \cup \bigcup_{i=1}^{p} x_i)$, $f'(u) = 0$ for all $u$ in $\bigcup_{i=1}^{p} x_i$ and $f'(v) = f(v) + \sum_{i=1}^{p} f(x_i)$.

We have both that $C_{f'}(V) = C_f(V)$ and that $|V'_f| < |V_f|$. It remains to show that $f'$ is a broadcast domination. Thus we have to show that any vertex hearing $x_i$, $i \leq p$, when $f$ was used as broadcast domination can hear $v$ now that $f'$ is used. Suppose $u$ heard $x_i$, $i \leq p$, when $f$ was used. As $x_i$ is a neighbor of $v$ in $G_f$, we have that $\delta(v,u) \leq \delta(v,x_i) + \delta(x_i,u) \leq f(v) + f(x_i) + 1 + f(x_i) \leq f(v) + f(x_p) + f(x_p) + 1$. But $f(x_p) + 1 \leq \sum_{i=1}^{p-1} f(x_i)$. Thus $f(v) + f(x_p) + f(x_p) + 1 \leq f(v) + \sum_{i=1}^{p} f(x_i) = f'(v)$ completing the proof. ∎

Let $f$ be a broadcast domination of $G$ satisfying the conditions in 6.2.1, and let $v$ be a vertex of degree $p > 2$ in the domination graph. Order the neighbors of $v$ in $G_f$ by power in nondecreasing order from $x_1$ to $x_p$. We know that $f(x_2 = \geq f(x_1) \geq 1$. Also $f(x_3) \geq f(x_2) + f(x_1) \geq 2$. If $p \geq 4$, $f(x_4) \geq f(x_3) + f(x_2) + f(x_1) \geq 4$ If $p \geq 5$, $f(x_5) \geq f(x_4) + f(x_3) + f(x_2) + f(x_1) \geq 8$. It is a simple exercise to use induction to prove that in fact, if $p = k \geq 2$ then $f(x_i) \geq 2^{k-2}$. As a consequence, $\sum_{i=1}^{k} f(x_i) \geq 2^{k-1}$. Since an optimal broadcast domination $f$ satisfies $n \geq C_f(V) \geq \sum_{i=1}^{k} f(x_i) \geq 2^{k-1}$ we have that $2n \geq 2^k = 2^p$. As $p$ was the degree of a vertex in $G_f$ we can conclude that $\Delta(G_f) \leq \log n + 1$.

**Observation 6.2.2** *If $f$ is an optimal broadcast domination of $G$ satisfying the conditions in Lemma 6.2.1 then* $\Delta(G_f) \leq \log n + 1$.

It seems difficult to exploit Lemma 6.2.1 to design an efficient algorithm for Weighted Broadcast Domination. At the same time, this result explains the difficulties with proving the problem $\mathcal{NP}$-complete, as it guarantees the existence of an optimal solution that is quite close to being efficient. Probably, stronger structural results are required in order to settle the computational complexity of the problem. Unfortunately, it seems very hard to squeeze out more information from arguments similar to that in the proof of Lemma 4.2.1, Theorem 4.4.2 or Lemma 6.2.1. For instance, one can show that the $\log n + 1$ bound on $\Delta(G_f)$ is asymptotically tight.

A *star* of size $k$ is a graph consisting of a vertex $u$, called the *central vertex*, with $k$ vertices of degree 1, *leaves*, attached to it. We define the *weighted star* $S_{k,w}$ to be a star of size $k$ with weight $w$ on all the edges. Observe that the optimal broadcast domination of $S_{k,k}$ is to let $f(u) = k$ for the central vertex $u$, and $f(l) = 0$ for all the leaves $l$.

Using stars we build a sequence of graphs $\{H_i\}$ for $i \geq 1$. We build the sequence inductively. The graph $H_0$ is a root vertex $r$ connected to the central vertex of a star $S_{2,1}$ with edges of weight 3. To build $H_{i+1}$ from $H_i$ you take the graph $H_i$ and attach an edge of weight $2^{i-1} + 2$ to the center of a new star $S_{2^{i-1}+1, 2^{i-1}}$.

The construction is the way it is so that optimal broadcast dominations of graphs in this sequence have a special form. The form of this broadcast domination is easy to

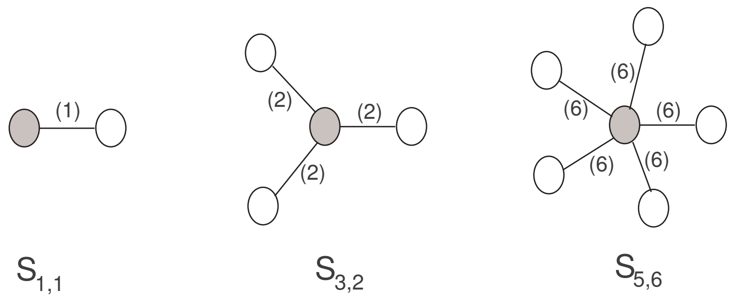


Figure 6.6: *The graphs $S_{1,1}$, $S_{3,2}$ and $S_{5,6}$*

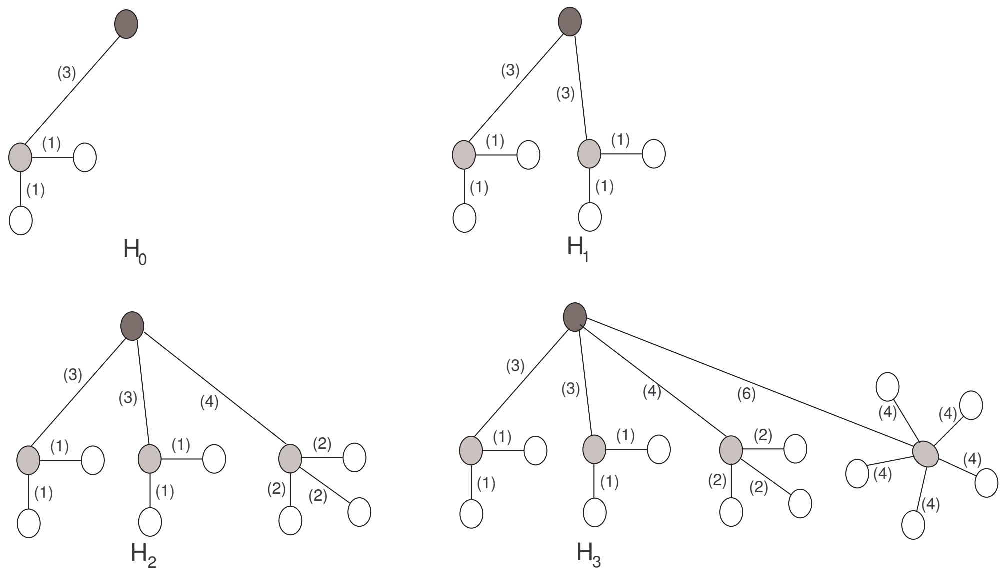


Figure 6.7: *The graphs $H_0$, $H_1$, $H_2$ and $H_3$. Root vertex is dark grey, central vertices of stars are light grey. Edge weights in parentheses.*

guess, but somewhat technically involved to prove.

**Lemma 6.2.3** *The only optimal broadcast domination $f$ of a graph $H_t$ satisfies $f(r) = 1$ for the root $r$, $f(l) = 0$ for all leaves, and $f(c) = w$ for every central vertex $c$ of a star $S_{k,w}$.*

**Proof.** We prove this by contradiction. Let $f$ be optimal and suppose the root vertex $r$ can hear a vertex $v$ in a star $S_{k,w}$ with central vertex $c$. Without loss of generality, assume that $v$ is the vertex of $S_{k,w}$ that maximizes $f(v) - \delta(v, r)$. Observe that in this case, $f(v) \geq w + 2$ Now, let $f'(u) = f(u)$ for all vertices $u$ not in the given star. Also let $f'(c) = w$ and $f(l) = 0$ for all leaves $l$ in the star, and let $f'(r) = f(r)+f(v)-w-2$. All vertices in the star now hear $c$. All vertices outside the star that heard $v$ now hear $r$. Hence $f'$ is a broadcast domination. The cost of $f'$ can be bounded from above, $C_{f'}(V) \leq C_f(V)-f(v)+w+f(v)-w-2 = C_f(V)-2$, contradicting that $f$ is optimal. Hence we can safely assume that in an optimal broadcast domination the root vertex cannot hear any other vertices and hence is a dominator.

If $f$ is optimal, all stars $S_{k,w}$ whose leaves that cannot hear the root vertex satisfy $f(c) = w$ for the central vertex and $f(l) = 0$ for all leaves of the star. Hence, if there is a vertex $v$ in one of the stars hearing the root vertex then $f(r) = \delta(r, l)$ for some leaf $l$. Thus $f(r) = 2^i + 2$ for a nonnegative integer $i$. In this case $C_f(V(H_t))$ is $2^i + 2 + \sum_{j=i+1}^{t} 2^{j-1} = 2 + 1 + \sum_{j=0}^{i-1} 2^j + \sum_{j=i}^{t-1} 2^j = 2 + 2^t$ which is larger than the cost of the broadcast domination proposed in the lemma, $2 + \sum_{j=i}^{t-1} 2^j = 1 + 2^t$. Hence, in an optimal broadcast domination of $H_t$ no vertices except the root vertex itself hears the root vertex. Thus $f(r) = 1$ and the result is proved. ■

Now, obviously $H_t$ has $\mathcal{O}(2^t)$ vertices while the degree of $r$ in the domination graph of the optimal broadcast domination is $t + 1$. Thus $\Delta(G_f) = \mathcal{O}(\log n)$ and the bound given in Observation 6.2.2 is asymptotically tight.

# Chapter 7

# Concluding remarks

In this thesis we have studied Broadcast Domination, and a couple variants of the problem. If we see Broadcast Domination as a generalization of Dominating Set it is somewhat surprising that while Dominating Set is $\mathcal{NP}$-complete, we were able to give an $\mathcal{O}(n^6)$ algorithm for Broadcast Domination. On one hand this is a textbook example that a small change in the problem statement of an $\mathcal{NP}$-complete problem often makes it polynomial time solvable. On the other hand, this change is usually a restriction on the input or output. In this case we have the opposite, Dominating Set is just Broadcast Domination with restricted output. From this we can draw the conclusion that while restrictions on the output narrow the search space they can rule out solutions that are easy to find. As an allegory, consider the task of finding a lion that is not in a city. The task might seem easier than that of just finding a lion, as we do not have to look in the cities. However, a simple way to find a lion is to go to the city zoo. Hence, we have made our life more difficult by restricting the search space.

In addition to Broadcast Domination, we considered Broadcast Domination with general cost functions. We showed that if the cost function $\zeta$ is a part of the input, Broadcast Domination is $\mathcal{NP}$-complete. Following this we proceeded to study Broadcast Domination with cost function $\zeta$ fixed to $\zeta(p) = p^2$, Quadratic Broadcast Domination. We conjectured that Quadratic Broadcast Domination is $\mathcal{NP}$-complete, and would be interested in a proof or counterexample for Conjecture 6.1.14.

Finally, we took a look at Weighted Broadcast Domination. It seems difficult to settle the computational complexity of this problem. Observation 6.2.2 should give an intuition why it is hard to give a proof of $\mathcal{NP}$-completeness, as it effectively bounds the amount of interdependencies in an optimal solution. At the same time Lemma 6.2.3 shows that we can not use the same approach for Weighted Broadcast Domination as for the original Broadcast Domination problem. Thus, any proof of $\mathcal{NP}$-completeness or algorithm for Weighted Broadcast Domination would probably have to contain some interesting new idea.

Also, while our $\mathcal{O}(n^6)$ algorithm for Broadcast Domination is polynomial, it is not fast enough to be able to run the algorithm on large data sets. Thus we would encourage the reader to attempt finding a more efficient algorithm for this problem.

# Bibliography


[1] J. Bar-Ilan, G. Kortsarz, and D. Peleg. How to allocate network centers. *J. Algorithms*, 15:385–415, 1993.

[2] C. Berge. *Theory of Graphs and its Applications.* Number 2 in Collection Universitaire de Mathematiques. Dunod, Paris, 1958.

[3] J. R. S. Blair, P. Heggernes, S. Horton, and F. Manne. Broadcast domination algorithms for interval graphs, series-parallel graphs, and trees. Technical Report 249, University of Bergen, Norway, 2003. Reports in Informatics.

[4] J. E. Dunbar, D. J. Erwin, T. W. Haynes, S. M. Hedetniemi, and S. T. Hedetniemi. Broadcasts in graphs. 2002. Submitted.

[5] D. J. Erwin. Dominating broadcasts in graphs. *Bull. Inst. Comb. Appl.*, 42:89–105, 2004.

[6] M. R. Garey and D. S. Johnson. *Computers and Intractability.* W. H. Freeman and Co., 1978.

[7] J. Guo, R. Niedermeier and D. Raible. *Improved Algorithms and Complexity Results for Power Domination in Graphs* Springer-Verlag LNCS 3623:172–184, 2005.

[8] T. W. Haynes, S. M. Hedetniemi, S. T. Hedetniemi and M. A. Henning. *Domination in Graphs Applied to Power Networks SIAM Journal of Disctete Mathematics* 15:519–529, 2002.

[9] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. *Domination in Graphs: Advanced Topics.* Marcel Dekker, New York, 1998.

[10] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. *Fundamentals of Domination in Graphs.* Marcel Dekker, New York, 1998.

[11] M. A. Henning. Distance domination in graphs. In T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, editors, *Domination in Graphs: Advanced Topics*, pages 321–349. Marcel Dekker, New York, 1998.

[12] P. Heggernes and D. Lokshtanov. Optimal broadcast domination of arbitrary graphs in polynomial time. In *Proceedings of the 31stWorkshop on Graph Theoretic Concepts in Computer Science- WG 2005*, pages 187 – 198. Springer Verlag, 2005. LNCS 3787.

[13] S. B. Horton, C. N. Meneses, A. Mukherjee, and M. E. Ulucakli. A computational study of the broadcast domination problem. Technical Report 2004-45, DIMACS Center for Discrete Mathematics and Theoretical Computer Science, 2004.

[14] T. H. Cormen, C. E. Leierson, R. L. Rivest, C. Steen. *Introduction To Algorithms, second edition.* McGraw-Hill, 2001.

[15] C. L. Liu. *Introduction to Combinatorial Mathematics.* McGraw-Hill, New York, 1968.

[16] O. Ore. *Theory of Graphs.* Number 38 in American Mathematical Society Publications. AMS, Providence, 1962.

[17] P. Sipser *Introduction to the Theory of Computation* PWS Publishing Company, Boston, 1997

[18] P. J. Slater. *R*-domination in graphs. *J. Assoc. Comput. Mach.*, 23:446–450, 1976.