Multipackings of Graphs

L. E. Teshima
Department of Mathematics and Statistics
University of Victoria, P. O. Box 3045, Victoria, BC
CANADA V8W 3P4
lteshima@uvic.ca

September 30, 2014

Abstract

A set $M \subseteq V$ is called a multipacking of a graph $G = (V, E)$ if, for each $v \in V$ and each $s$ such that $1 \leq s \leq \text{diam}(G)$, $v$ is within distance $s$ of at most $s$ vertices in $M$. The multipacking number, denoted $\text{mp}(G)$, is the maximum cardinality of a multipacking of $G$. A dominating broadcast of $G$ is a function $f : V \rightarrow \{0, 1, \ldots, \text{diam}(G)\}$ such that $f(v) \leq e(v)$ (the eccentricity of $v$) for all $v \in V$ and such that each vertex is within distance $f(v)$ from a vertex $v$ with $f(v) > 0$. The cost of a broadcast $f$ is $\sigma(f) = \sum_{v \in V} f(v)$, and the broadcast number $\gamma_b(G)$ is the minimum cost of a dominating broadcast. In this paper, we review a variety of recent results in the study of the dual graph parameters $\text{mp}$ and $\gamma_b$.

Keywords: Broadcast Domination, Multipackings, Fractional Multipackings, Farber’s Algorithm, Balanced Matrices

AMS Subject Classification 2010: 05C69, 05C70, 05C72 05C85, 90C27

1 Introduction

A broadcast on a connected graph $G = (V, E)$ is a function $f : V \rightarrow \{0, 1, \ldots, \text{diam}(G)\}$ such that $f(v) \leq e(v)$, for all $v \in V$, where $e(v)$ is the eccentricity of $v$. The set of broadcast vertices $V_f^+ = \{v : f(v) \geq 1\}$ is the set of vertices that transmit the broadcast. A vertex $v$ is said to hear a broadcast if there exists some broadcast vertex $u$ such that $d(u, v) \leq f(u)$. The $k$-neighbourhood of a vertex $v$ is the vertex subset $N_k[v] = \{u \in V : d(v, u) \leq k\}$. If $u$ is a broadcast vertex, $v$ hears the broadcast from $u$ if and only if $v$ is in the broadcast neighbourhood $N_f[u] = N_{f(u)}[u]$ of $u$. We say that $f$ is a dominating broadcast if each vertex of $G$ hears a broadcast. The cost of a broadcast $f$ is $\sigma(f) = \sum_{v \in V} f(v)$, and the broadcast number of $G$ is $\gamma_b(G)$, is the minimum cost of a dominating broadcast. A broadcast is efficient if each vertex hears exactly one broadcast. Conversely, a vertex is
said so be over-dominated if it hears multiple broadcasts. A dominating broadcast $f$ of $G$ such that $\sigma(f) = \gamma_b(G)$ is called a $\gamma_b$-broadcast.

If $f$ is a dominating broadcast such that $f(v) \in \{0, 1\}$ for each $v \in V$, then $\{v \in V : f(v) = 1\}$ is a dominating set of $G$; the smallest cardinality of a dominating set is the domination number $\gamma(G)$. A dominating subset $X \subseteq V(G)$ with $|X| = \gamma(G)$ is called a $\gamma$-set of $G$.

Broadcast domination was introduced as a generalization of ordinary domination by Erwin in his 2001 doctoral dissertation as cost domination [14] (see also [15]). Unlike ordinary domination, the minimum cost dominating broadcast problem can be solved in polynomial time for general graphs [22], and linear time for trees [10, 11]. This has made broadcasting a popular new research topic with many recent publications on broadcasts on trees [8, 13, 23, 24, 25, 29] and general graphs [1, 2, 3, 6, 12, 30].

In 2013, Brewster, Mynhardt and Teshima [6, 30] examined the minimum dominating broadcast problem as an integer programming (IP) problem. The resulting dual property was dubbed a multipacking. Formally, a vertex subset $M$ is a $k$-multipacking if for each $v \in V$ and each integer $s$ for $1 \leq s \leq k$, $|M \cap N_s[v]| \leq s$. The $k$-multipacking number $mp_k(G)$ is the maximum cardinality of a $k$-multipacking of $G$. When $k = \text{diam}(G)$, $M$ is called a multipacking and $mp_k(G)$ is the multipacking number, mp($G$).

Similarly to broadcasts and domination, multipackings are a generalization of 2-packings. A vertex subset $Y$ of a graph $G = (V, E)$ is a 2-packing if for each $v \in V$, $|Y \cap N[v]| \leq 1$. Thus, a 2-packing is a 1-multipacking. The 2-packing number $\rho(G)$ is the size of a maximum 2-packing of $G$.

Concepts not defined here can be found in [7, 9, 20, 21].

2 Broadcasts and Multipackings

2.1 Integer Programming Formulation

A dominating broadcast on a graph $G$ can also been viewed as a covering of $G$ with $k$-neighbourhoods centred at each broadcast vertex $v$ and each $k \in \{1, 2, \ldots, e(v)\}$. Thus a dominating broadcast can be seen as a collection of balls $B = \{N_k[v]\}$ such that for each $u \in V$ there exists some $N_k[v] \in B$ with $u \in N_k[v]$.

Suppose $G = (V, E)$ is a graph with $V = \{v_1, v_2, \ldots, v_n\}$. Let $c$, indexed by $(v, k)$ (where $v \in V$ and $1 \leq k \leq e(v)$), be the cost vector for the IP, and set each $c_{v,k} = k$. Furthermore, define the vector $x$, also indexed by $(v, k)$, so that each $x_{v,k}$ is an indicator variable in the IP’s solution. That is, for the optimal broadcast $f$ found by the IP,

$$x_{v,k} = \begin{cases} 1 & \text{if } f(v) = k \\ 0 & \text{otherwise.} \end{cases}$$

Finally, let $A$ be the incidence matrix with its $n$ rows indexed by the vertices $v_i$, and its
columns indexed by the pairs \((j,k)\), representing the \(k\)-neighbourhood of the vertex \(v_j\). The entries of \(A\) are therefore,

\[
a_{i,(j,k)} = \begin{cases} 
1 & \text{if } v_i \in N_k[v_j], \\
0 & \text{otherwise.}
\end{cases}
\]

We call \(A\) the extended neighbourhood matrix of \(G\). Thus the primal integer program (PIP) for a minimum cost broadcast, and the dual integer program (DIP) for a maximum multipacking, are as below.

**The Broadcast PIP:** \(B - PIP(G)\):

\[
\begin{align*}
\min & \quad c \cdot x \\
\text{s.t.} & \quad Ax \geq 1 \\
& \quad x_{k,v} \in \{0, 1\}.
\end{align*}
\]

**The Multipacking DIP:** \(MP - DIP(G)\):

\[
\begin{align*}
\max & \quad y \cdot 1 \\
\text{s.t.} & \quad yA \leq c \\
& \quad y_u \in \{0, 1\}.
\end{align*}
\]

Furthermore by the strong duality theorem of linear programs, we can concluded that for any graph \(G\),

\[
mp(G) \leq \gamma_b(G).
\]

### 2.2 Bounds, Differences and Ratios

In this section we present some recent results in comparing \(\gamma_b\) and \(mp\). We begin with one of the first results by Erwin, the bound presented in Proposition 1.

**Proposition 1** [14] For every non trivial connected graph \(G\),

\[
\left\lceil \frac{\text{diam}(G) + 1}{3} \right\rceil \leq \gamma_b(G) \leq \min\{\text{rad}(G), \gamma(G)\}.
\]

Hartnell and Mynhardt [19] have expanded this result to include multipackings.

**Proposition 2** [19], [30] For any connected graph \(G\),

\[
\left\lceil \frac{\text{diam}(G) + 1}{3} \right\rceil \leq mp(G) \leq \gamma_b(G) \leq \min\{\text{rad}(G), \gamma(G)\}.
\]
Let $d = \text{diam}(G)$ and $P = v_0, v_1, \ldots, v_d$ be a diametrical path of $G$. Define $V_i = \{v : d(v, v_0) = i\}$ and $M = \{v_i : i \equiv 0 \pmod{3}, i = 0, \ldots, d\}$. By our choice of $M$, any $v_i \in V_i$ satisfies $|N_s[v_i] \cap M| \leq s$ for all $s \geq 1$. Consider now any $1 \leq r \leq d$ and any $v \in V_r$. Since $v_r \in V_r$ is on $P$ and $M \subseteq V(P)$, $N_s[v] \cap M \subseteq N_s[v_r] \cap M$. Thus $|N_s[v] \cap M| \leq s$ for all $s \geq 1$. It follows that $\text{mp}(G) \geq |M| = \left\lceil \frac{\text{diam}(G) + 1}{3} \right\rceil$.

By combining the results of Propositions 1 and 2, Hartnell and Mynhardt closed an open problem from [6] which asked whether the $\gamma_b/\text{mp}$ ratio could be arbitrary.

**Corollary 3** [19] For any graph $G$, $$\frac{\gamma_b(G)}{\text{mp}(G)} < 3.$$  

**Proof.** Since $3 \text{mp}(G) \geq \text{diam}(G) + 1 > \text{rad}(G) \geq \gamma_b(G)$, $$\gamma_b(G) \leq 3 \text{mp}(G) - 1,$$ and so $$\frac{\gamma_b(G)}{\text{mp}(G)} < 3.$$ 

Hartnell and Mynhardt also offer the following result, which is the only known upper bound for $\gamma_b$ in terms of $\text{mp}$.

**Proposition 4** [19] If $G$ is a graph with $\text{mp}(G) \geq 2$, then $\gamma_b(G) \leq 3 \text{mp}(G) - 1$. Furthermore, equality is reached for some graphs $G$ with $\text{mp}(G) = 2$.

Naturally, the study of bounds has journeyed into investigations of equally. In particular, for which graphs is $\gamma_b = \text{mp}$? Trivially, $\gamma_b(P_n) = \text{mp}(P_n)$, where $P_n$ is the path on $n$ vertices. A similar result follows for cycles.

**Proposition 5** [30] For any cycle $C_n$ with $n \geq 3$, $\text{mp}(C_n) = \gamma_b(C_n)$ if and only if $n \equiv 0 \pmod{3}$.

A famous result in domination is the equality of $\gamma(T)$ and $\rho(T)$ for any tree, as shown by Meir and Moon in [26]. In 2013, Mynhardt and Teshima extended this result for broadcasts and multipackings.

**Theorem 6** [27] For any tree $T$, $\gamma_b(T) = \text{mp}(T)$.

The original proof for Theorem 6 provided in [27] is quite long and technical; however, it does provide a useful algorithm for finding a maximum multipacking of tree, which we present with example in Appendix A.2. Instead, we show Brewster and Duchesne’s alternative proof using Farber’s algorithm in Section 4.

Exploration has also ventured into examination of the $\gamma_b - \text{mp}$ gap. To start, we present the following proposition which provides a trivial condition for inequality between $\text{mp}$ and $\gamma_b$.
Proposition 7 [30] If $G$ has $\gamma(G) = \gamma_b(G)$ and does not have an efficient $\gamma$-set, then $\gamma_b - \text{mp} \geq 1$.

Figure 1: A graph $G$ with $\gamma_b(G) = 4$, $\text{mp}(G) = 2$.

The graph in Figure 1 (see [30]) was the first known example where $\gamma_b - \text{mp} > 1$. The following year, Hartnell and Mynhardt [19] provided a construction for a graph $G_k$ with $\gamma_b(G_k) - \text{mp}(G_k) = k$ for any $k \geq 1$, thereby demonstrating that the $\gamma_b - \text{mp}$ difference can be arbitrary.

Figure 2: The graph $G_1$ with $\gamma_b(G_1) = 4$ and $\text{mp}(G_1) = 3$

Construction of $G_k$ [19]: For $i = 1, 2, \ldots, 3k$, let $H_i \cong K_{2,4}$ with bipartition $(X_i, Y_i)$, where $X_i = \{c_i, u_i\}$ and $Y_i = \{a_i, r_i, s_i, b_i\}$. Form $G_k$ by joining $b_i$ and $a_{i+1}$ for each $i = 1, 2, \ldots, 3k - 1$.

The graph $G_1$ is illustrated in Figure 2. Notice that each induced subgraph $H_i$ contains at most one multipacking vertex, and so $\text{mp}(G_1) \leq 3$. Furthermore, recall from Proposition 2 that

$$\text{mp}(G_1) \geq \left\lceil \frac{\text{diam}(G_1) + 1}{3} \right\rceil = \left\lceil \frac{8 + 1}{3} \right\rceil = 3.$$

Thus the yellow vertex set (when viewed in colour) $U = \{u_1, u_2, u_3\}$ forms a maximum multipacking on $G_1$. The generalized graph $G_k$ is illustrated in Figure 3. The depicted broadcast $f$ with

$$f(v) = \begin{cases} 4 & \text{if } v = c_i \text{ and } i \equiv 2 \pmod{3} \\ 0 & \text{otherwise,} \end{cases}$$
Figure 3: The graph $G_k$ with $\gamma_b(G_k) = 4k$ and $mp(G_k) = 3k$

is clearly a dominating broadcast; we will show that $f$ is a minimum cost broadcast using Brewster and Duschesne’s application of fractional multipackings in Section 3.

The graph $G$ in Figure 1 has ratio $\gamma_b(G)/mp(G) = 2$, while the graph $G_k$ has $\gamma_b(G_k)/mp(G_k) = 4/3$. There are currently no known graphs with ratio $\gamma_b/ mp > 2$. Hartnell and Mynhardt note that if a graph $H$ has $mp(H) = 1$ or 2, then $\gamma_b(H)/mp(H) \leq 2$. Thus, if $G$ has $\gamma_b(G)/mp(G) > 2$, then $mp(G) \geq 3$. By Proposition 4, if $G$ has $mp(G) = 3$, then $\gamma_b(G) \leq 7$. It follows that if such a graph $G$ with $mp(G) = 3$ exists, then $G$ has $\gamma_b(G) = 7$. Hartnell and Mynhardt were unable to construct such an extremal graph; however, to aid in future investigation they formulated a series of seven structural facts.

**Facts [19]:** Suppose that $G$ is a connected graph with $mp(G) = 3$ and $\gamma_b(G) = 7$. Let $\alpha$ be a peripheral vertex of $G$. For $i = 0, 1, \ldots, 8$, let $V_i = \{x \in V(G) : d(x, \alpha) = i\}$.

(i) For all vertices $u, v \in V_3 \cup V_4$, $d(u, v) \leq 4$.
(ii) For any $u \in V_3 \cup V_4$, any $y \in V_5 \cup \cdots \cup V_8$ and any $w \in V_7 \cup V_8$, $d(u, y) \leq 4$ or $d(y, w) \leq 2$.
(iii) If $diam(G) = 8$, then for any $u \in V_3 \cup V_4$, $V_5 \subseteq N_4[u]$.
(iv) For each $u \in V_3 \cup V_4$, there exists $v \in V_4$ such that $d(u, v) \geq 3$. Furthermore, if $diam(G) = 7$, there is such a $v$ that $d(u, v) = 4$.
(v) Consider any $u, v \in V_3 \cup V_4$ such that $3 \leq d(u, v) \leq 4$.
   (a) If $diam(G) = 8$ and $\{u, v\} \cap V_3 \neq \emptyset$, then there exists a $u - v$ path of length at most four that contains a vertex in $V_2$.
   (b) If $diam(G) = 8$ and $\{u, v\} \subseteq V_3$, then there exists a $u - v$ path of length at most four that contains a vertex in $V_2$, or such a path that contains a vertex in $V_6$.
   (c) If $diam(G) = 7$ then there exist a $u - v$ path of length at most four that contains a vertex in $V_2$, or such a path that contains a vertex $V_5$.
(vi) For each $u \in V_3$ there exists $v \in V_2$ such that $d(u, v) = 5$. There also exists a path from $v$ to $V_3$ that does not contain $u$.
(vii) For each $u \in V_6 \cup V_7$ and each $w \in V_7$, $d(u, w) \leq 4$. Moreover, if $diam(G) = 8$, then for each $u \in V_6 \cup V_7 \cup V_8$ and each $w \in V_8$, $d(u, w) \leq 2$.

## 3 Fractional Broadcasts and Multipackings

Fractional relaxations are a natural extension of the broadcast and multipacking IP’s. The primal problem, the *fractional broadcast primal linear program* B-PLP, finds the minimum cost *fractional broadcast* of a graph $G$ with *fractional broadcast number* $\gamma_{b,f}(G)$. Now the
fractional value of $x_{k,v}$ can be viewed as the intensity or perhaps quality of the signal. For example $x_{k,v} = 1/2$ represents that half of a full signal is broadcast from vertex $v$ to all vertices at distance at most $k$ away. For a vertex $v$ to be dominated by a fractional broadcast, the sum of the signal intensities heard by $v$ must sum to at least one.

The Fractional Broadcast PLP: $B - PLP(G)$:

$$\begin{align*}
\min & \; c \cdot x \\
\text{s.t.} & \; Ax \geq 1 \\
& \; x_{k,v} \geq 0.
\end{align*}$$

For the fractional multipacking dual linear program MP-PLP, we view fractional multipackings as a weighting of the vertices rather a vertex subset. For a graph $G = (V, E)$, let $y_i$ be the multipacking weight of vertex $v_i$ and define $y$ to be the row vector with entries $y_i$.

The Fractional Multipacking DLP: $MP - DLP(G)$:

$$\begin{align*}
\max & \; y \cdot 1 \\
\text{s.t.} & \; yA \leq c \\
& \; y_u \geq 0.
\end{align*}$$

Again, by the strong duality theorem for linear programming,

$$mp \leq mp_f = \gamma_{b,f} \leq \gamma_b. \quad (1)$$

Fractional broadcasts have yet to be studied in any detail; however, by (1) it is possible that they will be useful in later investigation in determining which graphs have $mp = \gamma_b$. In the next section, we present some early results and applications of fractional multipackings, as investigated by Brewster and Duchesne [4].

3.1 Applications of Fractional Multipackings

Similar to the way multipackings can be used to certify the minimality of a given dominating broadcast, fractional multipackings can also be used as a certification tool in graphs where $mp < \gamma_b$. For example, recall the graph $G_k$ defined by Hartnell and Mynhardt in [19] and pictured in Figure 3 of Section 2. The given broadcast $f$ with $f(c_i) = 4$ for $i \equiv 2 \pmod{3}$, and $f(v) = 0$ otherwise is clearly dominating, but showing that $f$ is a minimum cost dominating broadcast is not immediate. The original proof by Hartnell and Mynhardt was fairly technical; Brewster and Duchesne offer a clever alternative.

Proposition 8 [19] The graph $G_k$ in Figure 3 has $\gamma_b(G_k) = 4k$.

Proof. Define a fractional multipacking $y$ on $G_k$ such that for each $i \in \{1, 2, \ldots, 3k\}$, $y_{r_i} = y_{s_i} = y_{c_i} = y_{u_i} = 1/3$ and $y_{a_i} = y_{b_i} = 0$. It is easy to see that for any vertex
\( v \in V(G_k) \) and \( 1 \leq \ell \leq 4, \)
\[ \sum_{u \in N_\ell[v]} y_u \leq \ell. \]
Furthermore for \( \ell \geq 5, \)
\[ \sum_{u \in N_\ell[v]} y_u \leq \sum_{u \in N_{\ell-3}[v]} y_u + \frac{8}{3} \leq \ell. \]
Therefore, \( y \) is a feasible fractional multipacking and by strong duality,
\[ \text{mp}_f(G_k) = y \cdot 1 = 4k = \gamma_b(G). \]

This immediately gives the following nice corollary.

**Corollary 9** [4] The mpf – mp difference can be arbitrarily large. The integrality gap is at most \( 3/4. \)

Brewster and Duchesne also investigated fractional multipackings on vertex transitive graphs. Let \( G \) be a vertex transitive graph and \( v \) any vertex of \( G \). Now let
\[ w_v(r) = \frac{r}{|N_r[v]|}. \]
Since \( G \) is vertex transitive, \( w_v(r) \) is the same for each vertex; let \( w(r) \) be this common value. Define
\[ w^* = \min_{1 \leq r \leq \text{rad}(G)} w(r) \]
and let \( r^* \) be the value of \( r \) such that \( w^* = w(r^*) \). For each \( r^* \)-neighbourhood of any vertex \( v \), let \( n = r^*/w^* \) be the number of vertices in \( N_{r^*}[v] \). By symmetry, if \( u \in N_{r^*}[v] \), then \( v \in N_{r^*}[u] \); therefore, each vertex belongs to exactly \( n \) \( r^* \)-neighbourhoods.

**Theorem 10** [4] For a vertex transitive graph \( G = (V, E) \), let \( y_v = w^* \) for all \( v \in V \). Then \( y \) is a maximum fractional multipacking.

**Proof.** Fix any \( u \in V \) and consider \( N_r[u] \) for some \( 1 \leq r \leq \text{rad}(G) \). Then, since
\[ \sum_{v \in N_r[u]} y_v = w^* \cdot |N_r[u]| = \frac{r^*}{|N_r[u]|} \cdot |N_r[u]| \leq r, \]
y is feasible. Now suppose that \( y' \) is any other feasible fractional multipacking of \( G \). Then
\[ y \cdot 1 = \sum_{u \in V} w^* = \sum_{u \in V} \frac{r^*}{n} \geq \sum_{u \in V} \frac{1}{n} \sum_{v \in N_{r^*}[u]} y'_v = \sum_{v \in V} \left( \sum_{u \in N_{r^*}[v]} \frac{1}{n} \right) = \sum_{v \in V} y'_v = y' \cdot 1. \]
Thus \( y \cdot 1 \geq y' \cdot 1 \) for any other fractional multipacking \( y' \), and therefore \( y \) is a maximum fractional multipacking.

For example, consider the Petersen graph \( P \). For any vertex \( v \in V(P) \),
\[
w(1) = \frac{1}{4} > \frac{2}{10} = \frac{1}{5} = w(2).
\]
Thus for the Petersen graph, \( w^* = 1/5 \), and \( mp_f(P) = 2 \). Notice that since \( \text{diam}(P) = \text{rad}(P) = 2 \), \( mp(P) = 1 \) and \( \gamma_b(P) = 2 \). Therefore, the Petersen graph provides another example where \( mp(P) < mp_f(P) = \gamma_b(G) \).

This notion of spreading the minimum fractional multipacking weight around a graph can also be used as a lower bound of \( mp_f \) for general graphs. For a graph \( G \), let
\[
w^* = \min_{v \in V, 1 \leq r \leq e(v)} w(v(r))
\]
and again set \( y_v = w^* \) for all \( v \in V \). Then \( |V| \cdot w^* \) is a trivial lower bound for \( mp_f \).

4 Farber’s Algorithm

In this section we examine a new perspective on broadcasts and multipackings currently being developed by Brewster and Duchesne in \cite{4}, and Brewster, MacGillivray and Yang in \cite{5}. This research provides an exciting amalgamation between the very young concept in multipackings and an older algorithm developed by Martin Farber in the early 1980’s. Since strongly chordal graphs play a pivotal role in the use of Farber’s algorithm, we begin this section with an introduction to strongly chordal graphs and some of their characterizations.

4.1 Strongly Chordal Graphs

A graph is chordal (or triangulated) if it does not contain an induced cycle of length greater than three. The class of chordal graphs contains many famous families including trees, threshold graphs, interval graphs, split graphs, and maximal outerplanar graphs. A graph \( G = (V, E) \) is said to have a perfect elimination ordering if its vertices can be ordered \( v_1, v_2, \ldots, v_n \) such that for each \( i, j \) and \( \ell \), if \( i < j, i < \ell \) and \( v_\ell, v_j \in N[v_i] \), then \( v_\ell \in N[v_j] \). In \cite{28}, Rose showed that a graph is chordal if and only if it has a perfect elimination ordering.

Farber \cite{16} strengthened this condition by defining a strong elimination ordering. A strong elimination ordering of a graph \( G = (V, E) \) is a vertex ordering \( v_1, v_2, \ldots, v_n \) such that for each \( i, j, k \) and \( \ell \), if \( i < j, k < \ell \) and \( v_k, v_\ell \in N[v_i] \) and \( v_k \in N[v_j] \), then \( v_\ell \in N[v_j] \). Farber defined a graph to be strongly chordal if it admits a strong elimination ordering. We examine three of Farber’s characterizations of strongly chordal graphs.

**Theorem 11** \cite{16} A graph is strongly chordal if and only if every induced subgraph of \( G \) contains some vertex \( v \) such that for each \( u, w \in N[v], N[u] \subset N[w] \) or vice versa; that is, \( G \) has a simple vertex \( v \).
To prove this result, Farber developed Algorithm 1 below (different from this section’s namesake) that when given any graph $G$ as input will either find a strong elimination ordering of $G$ or locate an induced subgraph of $G$ with no simple vertex. This algorithm is useful in its own right, as some algorithms (e.g. Algorithms 2 and 3) require a strong elimination ordering as input.

Algorithm 1

Input: A graph $G = (V, E)$.
Output: A strong elimination ordering or an induced subgraph without a simple vertex.
Initial: Set $n \rightarrow |V|$.

Step 1: Let $V_0 = V$ and let $(V_0, <_0)$ be a partial ordering on $V_0$ with $v <_0 u$ if and only if $v = u$. Let $V_1 = V$ and set $i \leftarrow 1$.

Step 2: Let $G_i$ be the subgraph of $G$ induced by $V_i$. If $G_i$ has no simple vertex, OUTPUT $G_i$ and STOP. Otherwise, define an ordering on $V_i$ by $v <_i u$ if $v <_{i-1} u$ or $N_i[v] \not\subset N_i[u]$.

Step 3: Choose a $v_i$ which is simple in $G_i$ and minimal in $(V_i, <_i)$. Let $V_{i+1} = V_i - \{v_i\}$. If $i = n$ OUTPUT ordering $v_1, v_2, \ldots, v_n$ of $V$ and STOP. Otherwise, set $i \leftarrow i + 1$ and GO TO Step 2.

Farber also determined a forbidden subgraph characterization for strongly chordal graphs. A trampoline is a split graph $G$ on $2n$ vertices for $n \geq 3$, with vertex partitions $W = \{w_1, w_2, \ldots, w_n\}$ and $U = \{u_1, u_2, \ldots, u_n\}$, where $W$ is independent, $G[U] \cong K_{\lvert U \rvert}$ and for each $i$ and $j$, $u_jw_i \in E(G)$ if and only if $i = j$ or $i \equiv j + 1(\text{mod } n)$. The trampolines on four and six vertices are illustrated in Figure 4 with the vertices of $U$ in blue and the vertices of $W$ in red, when viewed in colour.

Theorem 12

A chordal graph is strongly chordal if and only if it contains no induced trampoline subgraph.

The final characterization we present here is the most relevant to the study of multipackings, and graph optimization problems in general. For a graph $G$ with $V(G) = \{v_1, v_2, \ldots, v_n\}$, the neighbourhood matrix $M(G)$ is the $n \times n$ vertex-closed neighbourhood incidence matrix of $G$ with $m_{ij} = 1$ if $v_i \in N[v_j]$ and $m_{ij} = 0$ otherwise.

Proposition 13

For a graph $G$, the ordering $v_1, v_2, \ldots, v_n$ of its vertices is a strong elimination ordering if and only if the $\Gamma$-matrix,

$$
\Gamma = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix},
$$

is

is not a submatrix of the neighbourhood matrix $M(G)$.

This result is immediate by the definition of strong elimination ordering. Thus $G$ is strongly chordal if and only if $M(G)$ is $\Gamma$-free. A $(0,1)$-matrix is totally balanced if it does not contain an incidence matrix of any cycle of length at least three as a submatrix.

**Theorem 14** [16] A graph $G$ is strongly chordal if and only if $M(G)$ is totally balanced.

Farber’s proof is immediate by Proposition 13, observing that the $\Gamma$-matrix is an edge-vertex submatrix of every cycle of length at least three.

### 4.2 Farber’s Primal-Dual Algorithm

Farber’s original algorithm [17] is a linear-time search developed to find a minimum weight dominating set of a vertex subset of a strongly chordal graph. Following his notation, the weight of each vertex $v_i$ is denoted $w_i$. Since the problem of finding a minimum weight dominating set with arbitrary real weights can be reduced to the problem with real positive weights, we proceed with the assumption that $w_i > 0$ for all $i$. Furthermore, for vertices $v_i$ and $v_j$, if $i = j$ or $v_iv_j \in E(G)$, we write $i \sim j$. The definition of strong elimination ordering can be altered to utilize this new notation.

**Lemma 15** [17] For a graph $G$, an ordering $v_1, v_2, \ldots, v_n$ of its vertices is a strong elimination ordering if and only if for each $i, j, k, \ell$, with $i \leq j$ and $k \leq \ell$, and $i \sim k, i \sim \ell$ and $j \sim k$, then $j \sim \ell$.

We present a slight modification to Farber’s original LP problems here; for simplicity, in this paper we consider only the problem of finding a weighted dominating set of the entire graph, not just a vertex subset.

**The Primal** $P(G)$:

\[
\begin{align*}
\text{min} & \quad \sum_{i=1}^{n} w_i x_i \\
\text{s.t.} & \quad \sum_{i \sim j} x_i \geq 1 \text{ for each } j, \\
& \quad x_i \geq 0 \text{ for each } i.
\end{align*}
\]

**The Dual** $D(G)$:

\[
\begin{align*}
\text{max} & \quad \sum_{j=1}^{n} y_j \\
\text{s.t.} & \quad \sum_{j \sim i} y_j \leq w_i \text{ for each } i, \\
& \quad y_j \geq 0 \text{ for each } j.
\end{align*}
\]

11
Farber’s algorithm, as shown in Algorithm 2, solves the above LP \( P(\mathcal{G}) \) and its dual \( D(\mathcal{G}) \). When each vertex \( v_i \) of a graph \( \mathcal{G} \) has weight \( w_i = 1 \), this is equivalent to finding a \( \gamma \)-set in the primal problem and a \( \rho \)-set in the dual problem.

Algorithm 2 executes in two stages. In the first stage, it finds an optimal solution to the dual problem \( D(\mathcal{G}) \) by scanning the vertices in the given strong elimination order \( v_1, v_2, \ldots, v_n \). Here, it greedily adds vertices to the 2-packing by considering the available slackness in each associated neighbourhood of each vertex. In the second stage the algorithm finds an optimal solution to the primal \( P(\mathcal{G}) \) by scanning the closed neighbourhoods of vertices in reverse strong elimination order \( N[v_n], N[v_{n-1}], \ldots, N[v_1] \). This finds a dominating set by examining vertices whose neighborhoods have no remaining slack after Stage 1.

The set \( T \) is used to assure that the complementary slackness conditions are satisfied as the algorithm moves through Stage 2. Farber also defines

\[
h(i) = w_i - \sum_{j \sim i} y_j \tag{2}\]

and

\[
T_i = \{ k : k \sim i \text{ and } y_k > 0 \} \tag{3}\]

to track available slackness. When the algorithm begins, \( T = \{1, 2, \ldots, n\} \), \( x_i = 0 \) and \( y_j = 0 \); the 2-packing and dominating sets are empty, and thus every vertex has slack.

**Algorithm 2**

**Input:** A strongly chordal graph \( \mathcal{G} = (V, E) \) with strong elimination ordering \( v_1, v_2, \ldots, v_n \) and positive vertex weights \( w_1, w_2, \ldots, w_n \).

**Output:** Optimal solutions to \( P(\mathcal{G}) \) and \( D(\mathcal{G}) \).

**Initial:** Set \( T = \{1, 2, \ldots, n\} \) and each \( y_j = 0 \), \( x_i = 0 \).

**Step 1:** For each \( j = 1, \ldots, n \), set \( y_j \leftarrow \min \{ h(k) : k \sim j \} \).

**Step 2:** For each \( i = n, \ldots, 1 \), if \( h(i) = 0 \) and \( T_i \subset T \), then set \( x_i \leftarrow 1 \) and \( T \leftarrow T - T_i \).

The algorithm clearly halts in \( O(2n) \) operations. Step 1 ensures that \( y_j \geq 0 \) and \( h(j) \geq 0 \) for \( j \), and therefore the solution presented by Algorithm 2 is dual feasible. Furthermore, for each \( i \), \( x_i \in \{0, 1\} \). Thus to show the feasibility of the primal solution, it suffices to show that for each \( j \), there is some \( i \sim j \) with \( x_i = 1 \). Since \( y_j \) is the \( \min \{ h(k) : k \sim j \} \), there exists some \( k \sim j \) such that \( h(k) = 0 \) and \( \max T_k \leq j \). If \( x_k = 1 \), we are done. Otherwise, if \( x_k = 0 \), when the algorithm scanned \( v_k \) in Stage 2 \( T_k \) was not a subset of \( T \). Since the vertices in Stage 2 are scanned in descending index order, this implies that there is some \( \ell > k \) such that \( x_\ell = 1 \) and \( T_\ell \cap T_k \neq \emptyset \). Let \( i \in T_\ell \cap T_k \), and then by transitivity of \( \sim \), \( \ell \sim i \sim k \sim j \). It follows that \( i \leq j \) since \( \max T_k \leq j \). The vertices were presented in a strong elimination order, and so by Lemma 15 it follows that \( \ell \sim j \). Therefore, there exists some \( \ell \sim j \) with \( x_\ell = 1 \) as required. This demonstrates the feasibility of the primal solution.
To confirm that the solutions are optimal, suppose that some \( x_i > 0 \); that is, \( x_i = 1 \). Then there is no slackness available around \( v_i \), so \( h(i) = 0 \). By (2),
\[
\sum_{j \sim i} y_j = w_i.
\]
Now suppose \( y_j > 0 \). It follows that \( \sum_{i \sim j} x_i \leq 1 \) because the algorithm requires that if \( x_i = x_k = 1 \), then \( T_i \cap T_k = \emptyset \). Combining this with the feasibility requirement that \( \sum_{i \sim j} x_i \geq 1 \) yields that
\[
\sum_{i \sim j} x_i = 1.
\]
Thus both the primal and dual solution are tight and therefore optimal.

See Example 1 in Appendix A.1 for an example of Algorithm 2 applied to a tree.

4.3 Extension to Broadcasts and Multipackings

Recently, Brewster and Duchesne [4] extended Farber’s original algorithm (Algorithm 2) to broadcasts and multipackings. The original primal solution from the algorithm provided a minimum weight dominating set. To account for the farther reaching nature of broadcasts, Brewster and Duchesne modified the algorithm to instead search for a minimum weight covering of \( k \)-neighbourhoods (or balls), where each \( k \)-neighbourhood had weight (or cost) \( k \).

As Brewster and Duchesne’s paper is still being drafted, we have taken some liberties in guessing applicable notation and the exact formulation of Algorithm 3. Recall from Section 1 that for a graph \( G \) with \( V(G) = \{v_1, v_2, \ldots, v_n\} \), the extended neighbourhood matrix \( A \) is the \( n \times m \) vertex- multi-neighbourhood incidence matrix of \( G \). The \( n \) rows are indexed by the vertices \( v_i \) and the \( m \) columns are indexed as pairs \((j, k)\) to denote the \( k \)-neighbourhood of vertex \( v_j \). The entries of \( A \) are such that
\[
a_{i,(j,k)} = \begin{cases} 
1 & \text{if } v_i \in N_k[v_j], \\
0 & \text{otherwise.}
\end{cases}
\]
For convenience, we also extend some of Farber’s notation. We write \( i \sim_m (j, k) \) if either \( i = j \) or \( v_i \in N_k[v_j] \).

Notice that the weights are assigned to vertex neighbourhoods, rather than the vertices themselves. The weight of the \( k \)-neighbourhood of the vertex \( v_i \) is denoted \( w_{i,k} = k \). Thus, let
\[
h(i, k) = w_{i,k} - \sum_{j \sim_m (i, k)} y_j = k - \sum_{j \sim_m (i, k)} y_j,
\]
and
\[
T_{i,k} = \{ j : j \sim_m (i, k) \text{ and } y_j > 0 \}.
\]
Algorithm 3 [4]

Input: A weighted graph with $G = (V, E)$ with strong elimination order $v_1, v_2, \ldots, v_n$.

Output: Optimal solutions to BIP($G$) and MIP($G$).

Initial: Set $T = \{1, 2, \ldots, n\}$ and each $y_j = 0$ and $x_{i,k} = 0$.

Step 1: For each $j = 1, \ldots, n$, set $y_j \leftarrow \min\{h(i,k) : j \sim_m (i,k)\}$.

Step 2: For $(i,k)$ in descending lexicographic order, if $h(i,k) = 0$ and $T_{i,k} \subset T$, then set $x_{i,k} \leftarrow 1$ and $T \leftarrow T - T_{i,k}$.

For each $v_i$, we could consider each of its $k$-neighbourhoods for $k = 1, \ldots, e(v)$; however, in an optimal broadcast setting, we can ignore certain neighbourhoods that we know will never be selected in a minimum cost broadcast. For example, if $v_\ell$ is a leaf of a tree with $e(v_\ell) \geq 2$, there is no incentive to define a broadcast $f$ with $f(v_\ell) = e(v_\ell)$; if $v$ is the stem of $v_\ell$ we can always cover at least as many vertices with a broadcast $g(v) = e(v_\ell)$. Thus, we can safely remove some neighbourhoods from $A$. It is likely that exactly which neighbourhoods are removable is dependent upon each class of graph.

To apply the algorithm to a tree $T$, Brewster and Duchesne give the following construction of a specifically ordered extended neighbourhood matrix $M$. We provide an example of Algorithm 3 being applied to a tree in Example 2 in Appendix A.1.

Construction of $M$: Given any tree $T$, root $T$ at a central vertex. For each $v \in V(T)$, let $\ell(v)$ be the maximum distance to a leaf below $v$ in $T$. For each non-leaf vertex $v$, construct a series of balls of radius $1, 2, \ldots, \ell(v)$ centred at $v$. Define $M$ to be the resulting vertex-ball incidence matrix, with $n$ rows sorted in descending (rooted) level order, and $m$ columns sorted left to right in ascending lexicographic order read from the bottom up.

Proposition 16 [4] The resulting matrix $M$ is $\Gamma$-free.

Proof. Suppose to the contrary that $M$ contains a $\Gamma$-submatrix. Then there exist two balls $B$ and $A$, and two vertices $z$ and $y$ such that $z \in A \cap B$, $y \in B$ and $y \notin A$, as shown in the vertex-expanded neighbourhood submatrix below. Since the columns of $M$ are sorted lexicographically, there exists some other vertex $x$, such that $x \in A$ but $x \notin B$; otherwise, the column representing $A$ would be to the left of the column representing $B$.

|   | B | A |
|---|---|---|
| z | 1 | 1 |
| y | 1 | 0 |
| x | 0 | 1 |

Let $a$ be the centre vertex of ball $A$, $b$ be the centre vertex of ball $B$, and $w$ be the least common ancestor of $z$ and $y$.

Case 1: $w$ is on the $z - x$ path. Then $d(z, x) = d(z, w) + d(w, x)$. Since the rows were sorted by decreasing depth, the depth of $z$ in $T$ is at least that of $x$ and $y$. Hence $d(w, y) \leq d(w, z)$. Notice that $a$ is not the $y - w$ path, since otherwise $d(a, y) \leq d(a, z)$ which implies that
$y \in A$. Suppose that $w$ is on the $a - z$ path. Then

$$d(a, y) = d(a, w) + d(w, y) \leq d(a, w) + d(w, z) = d(a, z).$$

Since $z \in A$ and $d(a, y) \leq d(a, z)$, this implies that $y \in A$, a contradiction. Thus $a$ and $z$ share the same child of $w$ as an ancestor, and $w$ is on both the $a - x$ and $a - y$ paths. Recall that $x \in A$ and $y \notin A$, and so $d(a, x) < d(a, y)$, which implies

$$d(a, w) + d(w, x) = d(a, x) < d(a, y) = d(a, w) + d(w, y).$$

It follows that $d(w, x) < d(w, y) \leq d(w, z)$.

By a similar argument (substituting $b$ and $x$ for $a$ and $y$, respectively), we conclude that $w$ is not on the $b - z$ path, but is on the $b - x$ and $b - y$ paths. Thus,

$$d(b, x) = d(b, w) + d(w, x) < d(b, w) + d(w, y) = d(b, y).$$

Finally, since $d(b, x) < d(b, y)$ and $y \in B$, it follows that $x \in B$, a contradiction.

**Case 2:** $w$ is not on the $z - x$ path. Let $v$ be the lowest common ancestor of $x$ and $z$. A similar argument to Case 1 using $v$ in place of $w$ again implies that $x \in B$ and forms the desired contradiction.

Forthcoming work by Brewster and Duchesne will demonstrate how this result provides a nice alternative proof to Theorem 6. Furthermore, Proposition 16 demonstrates that both a minimum broadcast and a maximum multipacking can be found in $O(n + m)$ time, where $n$ is the number of vertices and $m$ is the width of the matrix $M$ constructed above Proposition 16.

In a currently unpublished work, Brewster, MacGillivray and Yang [5] extend this result to show that the extended neighbourhood matrix of a graph $G$ is $\Gamma$-free if and only if $G$ is strongly chordal. Although this implies that Algorithm 3 can only be applied to strongly chordal graphs, this does not complete the class of graphs with $\gamma_b = mp$. Recall for example that $\gamma_b(C_6) = mp(C_6)$, but $C_6$ is not chordal.

5 Conclusions

Having examined some of the main results in the very young study of multipackings and broadcasts in graphs, we conclude our survey with some open problems.

5.1 Open Problems

**Problem 1** [6] For which graphs $G$ is $\gamma_b(G) = mp(G)$?

**Problem 2** [19] Does there exist a graph $G$ with $mp(G) = 3$ and $\gamma_b(G) \geq 5$?

**Problem 3** [19] Can the ratio $\gamma_b / mp < 3$ be improved for general graphs?
Problem 4 Does there exist a graph $G$ with integral $\gamma_{b,f}(G)$ such that

$$mp(G) < \gamma_{b,f}(G) < \gamma_b(G)?$$

Problem 5 In [17], Farber modified his algorithm to find minimum independent dominating sets of strongly chordal graphs. Dunbar et al. [12] define a broadcast $f$ to be independent if for each $v \in V_f^+$, $N_f[v] \cap V_f^+ = \{v\}$, or equivalently $|\{u \in V_f^+ : d(u,v) \leq f(u)\}| = 1$. Thus if a broadcast is independent, then each broadcast vertex hears only the broadcast from itself. Can Farber’s algorithm for independent dominating sets be extended to independent dominating broadcasts? Furthermore, what is the dual parameter to the independent broadcast number?

Problem 6 Can the minimum cost broadcast and maximum multipacking problems be formulated as hypergraph transversal and matching problems?

Problem 7 A clutter is a hypergraph with no nested edges. As detailed in [9], clutters have been extensively researched in an optimization context. In general, the extended neighbourhood hypergraph is not a clutter, since it has many nested edges; however, the hypergraph whose edges are the broadcast neighbourhood of an efficient broadcast is a trivial clutter. Is there a meaningful way to interpret the minimum cost broadcast or maximum multipacking problem in terms of clutters?

A Examples

A.1 Farber’s Algorithm

Example 1. Using Farber’s original algorithm (Algorithm 2), we find a maximum $2$-packing and minimum dominating set for the graph $G$ in Figure 5.

Figure 5: A tree $G$ with strong elimination order $v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}$. 
Table 1: The neighbourhood matrix of $T$

**Initial:** $x_v = y_v = 0$ for all $v \in V(G)$, $T = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.

| $v_1$ | $N[v_1]$ | $N[v_2]$ | $N[v_3]$ | $N[v_4]$ | $N[v_5]$ | $N[v_6]$ | $N[v_7]$ | $N[v_8]$ | $N[v_9]$ | $N[v_{10}]$ |
|-------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-------------|
| 1     | 0         | 0         | 1         | 0         | 0         | 0         | 0         | 0         | 0         | 0           |
| 2     | 0         | 1         | 0         | 0         | 1         | 0         | 0         | 0         | 0         | 0           |
| 3     | 0         | 0         | 1         | 0         | 0         | 1         | 0         | 0         | 0         | 0           |
| 4     | 0         | 0         | 0         | 1         | 0         | 0         | 1         | 0         | 0         | 0           |
| 5     | 0         | 0         | 0         | 0         | 1         | 0         | 0         | 0         | 0         | 0           |
| 6     | 0         | 0         | 0         | 0         | 0         | 1         | 0         | 1         | 0         | 0           |
| 7     | 0         | 0         | 0         | 1         | 1         | 0         | 1         | 0         | 1         | 0           |
| 8     | 0         | 0         | 0         | 0         | 0         | 1         | 0         | 1         | 0         | 1           |
| 9     | 0         | 0         | 0         | 0         | 0         | 0         | 1         | 0         | 1         | 1           |
| 10    | 0         | 0         | 0         | 0         | 0         | 1         | 1         | 1         | 1         | 1           |

**Stage 1:** Scan rows in descending order.

- $h(i) > 0$ for all $i \sim 1$.
  
  **Update:** $y_1 = 1$.

| $v_i$ | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ | $v_6$ | $v_7$ | $v_8$ | $v_9$ | $v_{10}$ |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-----------|
| $h(i)$ | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1         |

- $h(i) > 0$ for all $i \sim 2$.
  
  **Update:** $y_2 = 1$.

| $v_i$ | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ | $v_6$ | $v_7$ | $v_8$ | $v_9$ | $v_{10}$ |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-----------|
| $h(i)$ | 0     | 1     | 1     | 0     | 1     | 1     | 1     | 1     | 1     | 1         |

- $h(i) > 0$ for all $i \sim 3$.
  
  **Update:** $y_3 = 1$.

| $v_i$ | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ | $v_6$ | $v_7$ | $v_8$ | $v_9$ | $v_{10}$ |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-----------|
| $h(i)$ | 0     | 0     | 1     | 0     | 1     | 1     | 1     | 1     | 1     | 1         |

- $h(4) = 0$ and $4 \sim 4$.
  
  **Keep:** $y_4 = 0$.

- *Similarly for $v_5, v_6, v_7, v_8$.**
• $h(i) > 0$ for all $i \sim 9$.

Update: $y_9 = 1$.

| $v_i$ | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ | $v_6$ | $v_7$ | $v_8$ | $v_9$ | $v_{10}$ |
|------|------|------|------|------|------|------|------|------|------|--------|
| $h(i)$ | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

• $h(10) = 0$ and $10 \sim 10$.

Keep: $y_{10} = 0$.

• STOP: All vertices scanned.

**Stage 2:** Scan neighbourhoods in reverse order.

• $h(10) = 0$ and $T_{10} = \{9\} \subseteq T$.

Update: $x_{10} = 1$ and $T = \{1, 2, 3, 4, 5, 6, 7, 8, 10\}$.

• $h(9) = 0$ but $T_9 = \{9\} \not\subseteq T$.

Keep: $x_9 = 0$.

• $h(8) = 1$. Keep: $x_8 = 0$.

• $h(7) = 0$ but $T_7 = \{9\} \not\subseteq T$.

Keep: $x_7 = 0$.

• $h(6) = 0$ and $T_6 = \{3\} \subseteq T$.

Update: $x_6 = 1$ and $T = \{1, 2, 4, 5, 6, 7, 8, 10\}$.

• $h(5) = 0$ and $T_5 = \{2\} \subseteq T$.

Update: $x_5 = 1$ and $T = \{1, 4, 5, 6, 7, 8, 10\}$.

• $h(4) = 0$ and $T_4 = \{1\} \subseteq T$.

Update: $x_4 = 1$ and $T = \{4, 5, 6, 7, 8, 10\}$.

• STOP: $\sum_{v_i \in V} y_i = \sum_{v_i \in V} x_i$

Therefore $\{1, 2, 3, 9\}$ is a maximum 2-packing and $\{4, 5, 6, 10\}$ is a minimum dominating set of $G$.

**Example 2.** Using Brewster and Duchesne’s modification of Farber’s algorithm (Algorithm 3), we find a maximum multipacking and minimum dominating broadcast for the graph $G$ in Figure 5. For space, we use the notation $(i, k)$ in place of $N_k[v_i]$.
\[
N_k[v_i] \begin{array}{cccccccccccccccc}
(1,1) & (2,1) & (3,1) & (4,1) & (5,1) & (6,1) & (7,1) & (8,1) & (9,1) & (7,2) & (10,1) & (8,2) & (9,2) & (10,2) & (9,3) & (10,3) & (10,4) \\
v_1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
v_2 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
v_3 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
v_4 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
v_5 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\
v_6 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
v_7 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
v_8 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
v_9 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
v_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{array}
\]

Table 2: The extended neighbourhood matrix of \( T \)

**Initial:** \( y_i = 0 \) and \( x_{i,k} = 0 \) for all \( v_i \in V(G) \)

\( T = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \).

\[
N_k[v_i] \begin{array}{cccccccccccccccc}
(1,1) & (2,1) & (3,1) & (4,1) & (5,1) & (6,1) & (7,1) & (8,1) & (9,1) & (7,2) & (10,1) & (8,2) & (9,2) & (10,2) & (9,3) & (10,3) & (10,4) \\
h(i, k) & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 2 & 2 & 2 & 3 & 3 & 4 \\
\end{array}
\]

**Stage 1:** Scan rows in descending order.

- \( h(i, k) > 0 \) for all \( h(i, k) \) such that \( 1 \sim_m (i, k) \).
  
  Update: \( y_1 = 1 \).

\[
N_k[v_i] \begin{array}{cccccccccccccccc}
(1,1) & (2,1) & (3,1) & (4,1) & (5,1) & (6,1) & (7,1) & (8,1) & (9,1) & (7,2) & (10,1) & (8,2) & (9,2) & (10,2) & (9,3) & (10,3) & (10,4) \\
h(i, k) & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 \\
\end{array}
\]

- \( h(i, k) > 0 \) for all \( h(i, k) \) such that \( 2 \sim_m (i, k) \).
  
  Update: \( y_2 = 1 \).

\[
N_k[v_i] \begin{array}{cccccccccccccccc}
(1,1) & (2,1) & (3,1) & (4,1) & (5,1) & (6,1) & (7,1) & (8,1) & (9,1) & (7,2) & (10,1) & (8,2) & (9,2) & (10,2) & (9,3) & (10,3) & (10,4) \\
h(i, k) & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 2 & 2 & 2 & 1 & 3 & 2 \\
\end{array}
\]

- \( h(i, k) > 0 \) for all \( h(i, k) \) such that \( 2 \sim_m (i, k) \).
  
  Update: \( y_3 = 1 \).

\[
N_k[v_i] \begin{array}{cccccccccccccccc}
(1,1) & (2,1) & (3,1) & (4,1) & (5,1) & (6,1) & (7,1) & (8,1) & (9,1) & (7,2) & (10,1) & (8,2) & (9,2) & (10,2) & (9,3) & (10,3) & (10,4) \\
h(i, k) & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 2 & 2 & 1 & 2 & 1 \\
\end{array}
\]

- \( h(1, 1) = 0 \)
  
  Keep: \( y_4 = 0 \).
Similarly for 5, 6, 7, 8, 9, 10.

STOP: All vertices scanned.

Stage 2: Scan neighbourhoods in reverse order.

• \( h(10, 4) > 0 \). Keep \( x_{10,4} = 0 \).

• Similarly for \( (10, 3), (9, 3), (10, 2), (9, 2), (8, 2), (10, 1) \).

• \( h(7, 2) = 0 \) and \( T_{7, 2} = \{1, 2\} \subseteq T \).
  Update: \( x_{7,2} = 1 \) and \( T = \{3, 4, 5, 6, 7, 8, 9, 10\} \).

• \( h(9, 1) > 0 \). Keep \( x_{9,1} = 0 \).

• Similarly for \( (8, 1), (7, 1) \).

• \( h(6, 1) = 0 \) and \( T_{6, 1} = \{3\} \subseteq T \).
  Update: \( x_{6,1} = 1 \) and \( T = \{4, 5, 6, 7, 8, 9, 10\} \).

• STOP: \( \sum_{v_i \in V} y_i = \sum_{v_i \in V,k} x_{i,k} \)

Therefore \( \{v_1, v_2, v_3\} \) is a maximum 2-packing and the broadcast

\[
f(v_i) = \begin{cases} 
2 & \text{for } i = 7 \\
1 & \text{for } i = 6 \\
0 & \text{otherwise}, 
\end{cases}
\]

is a minimum dominating broadcast of \( G \).

A.2 The Tree Multipacking Algorithm

In this section, we provide the original algorithm from [30] for finding a maximum multipacking of a tree. Before presenting said algorithm, we first supply the necessary definitions and notations.

Let \( P : v_0, ..., v_d \) be a diametrical path of a tree \( T \) with \( \text{diam}(T) = d \). For each \( v_i \in V(P) \), let \( U_i \) be the set of all vertices of \( T \) that are connected to \( v_i \) by a (possibly trivial) path internally disjoint from \( P \). Let \( u_i \) be a vertex in \( U_i \) at maximum distance from \( v_i \), and let \( B_i \) be the \( v_i - u_i \) path. The shadow tree \( S_{T,P} \) of \( T \) with respect to \( P \) is the subtree of \( T \) induced by \( \bigcup_{i=0}^{d} V(B_i) \). If \( T \cong S_{T,P} \) for some diametrical path \( P \) of \( T \), then \( T \) is also called a shadow tree. Note that a shadow tree has maximum degree at most three.

Consider a shadow tree \( S_{T,P} \). If \( U_i - \{v_i\} = \emptyset \), we call \( v_i \) a branch vertex and the \( v_i - u_i \) path \( B_i \) a branch. Furthermore, for \( \alpha_i = d(v_i, u_i) \geq 1 \), the tree \( \Delta_i \) induced by \( \{v_i-\alpha_i, \ldots, v_i-1\} \cup V(B_i) \cup \{v_i+1, \ldots, v_i+\alpha_i\} \) is called the triangle at \( i \). If the vertex subset \( \{v_i-\alpha_i, \ldots, v_i, \ldots, v_i+\alpha_i\} \) of the triangle \( \Delta_i \) is contained in the vertex subset
\{v_{j-α}, ..., v_j, ..., v_{j+α}\} \text{ of the triangle } Δ_j, \text{ then } Δ_i \text{ is called a } nested \text{ triangle. A free edge is an edge of } S_{T,P} \text{ that is not in any triangle; note that all free edges of } S_{T,P} \text{ lie on } P.

The triangles of } S_{T,P} \text{ are labeled in order of their occurrence on } P \text{ and are denoted } Δ_{i_1}, Δ_{i_2}, ..., Δ_{i_c}. \text{ For simplicity, we abuse notation and denote } Δ_{i_1} \text{ as } Δ_1 \text{; and } Δ_{i_c} \text{ as } Δ_c. \text{ A free edge on } P \text{ that comes before } Δ_1 \text{ is called a leading free edge; likewise, a free edge that comes after } Δ_c \text{ is called a trailing free edge. If } e \text{ is a free edge of } S_{T,P}, \text{ we also call } e \text{ a free edge of } T \text{ with respect to } P. \text{ A set } M \text{ of edges of the diametrical path } P \text{ of the tree } T \text{ is a split-P set if each component } T' \text{ of } T - M \text{ has a positive even diameter and } P' = T' \cap P \text{ is a diametrical path of } T'. \text{ A split-set of } T \text{ is a split-P set for some diametrical path } P \text{ of } T. \text{ An edge in any split-set of } T \text{ is a split-edge. The requirement that } P' = T' \cap P \text{ be a diametrical path of } T' \text{ implies that each split-edge is a free edge. However, not all free edges are split-edges.}

![Diagram](https://example.com/diagram.png)

Figure 6: A labelled shadow tree } S_{T,P}.

We illustrate the following notation in Figure 6. Let } c \text{ be the highest index such that } v_c \text{ is a branch vertex of } T. \text{ The subpath } Q_c : v_c, ..., v_d \text{ of } P \text{ is called the trailing endpath of } T. \text{ The branch of } T \text{ that starts at } v_c \text{ is the path } B_c : v_c = u_{c,0}, u_{c,1}, ..., u_{c,α} \text{ of length } α, \text{ and is called the last branch of } T. \text{ The triangle } Δ_c \text{ associated with } B_c \text{ is called the last triangle of } T. \text{ For brevity we also write } B_c \text{ and } Q_c \text{ for } V(B_c) \text{ and } V(Q_c), \text{ respectively. We denote the lengths of } B_c \text{ and } Q_c \text{ by } ℓ(B_c) \text{ and } ℓ(Q_c), \text{ respectively; note that } α = ℓ(B_c) \leq ℓ(Q_c) = d - c.
The first and last edges of $\Delta_c$ on $P$ are $e_f = v_{c-\alpha}v_{c-\alpha+1}$ and $e_\ell = v_{c+\alpha-1}v_{c+\alpha}$, respectively.

Algorithm 4: **FindTreeMP** finds a maximum multipacking of a tree

_**Input:**_ Shadow tree $T$ with no nested triangles, diametrical path $P = \{v_0, v_1, \ldots v_d\}$

_**Output:**_ A maximum multipacking $M$ of $T$

$M \leftarrow \emptyset$

$S \leftarrow \emptyset$

while $T \neq P_1, P_2, P_3$ do

if $c = i_1$ and $l(B_c) > l(Q_1)$ then

$P \leftarrow P - Q_1 + B_c$

comment: $Q_1$ becomes $B_c$

end

if $\Delta_{ij}$ is a nested triangle then

$T \leftarrow T - (B_{ij} - \{v_{ij}\})$

end

if number of trailing free edges $\geq 3$ then

$M \leftarrow M \cup \{v_d\}$

$P \leftarrow P - \{v_d, v_{d-1}, v_{d-2}\}$

$T \leftarrow T - \{v_d, v_{d-1}, v_{d-2}\}$

else if number of trailing free edges $= 2$ then

$M \leftarrow M \cup \{v_d\}$

$P \leftarrow P - \{v_d, v_{d-1}, v_{d-2}\}$

$T \leftarrow T - \{v_d, v_{d-1}, v_{d-2}\}$

$P \leftarrow P - Q_c + B_c$

comment: $Q_c$ becomes $B_c$

else if number of trailing free edges $= 1$ then

$S \leftarrow S \cup \{(v_c, v_{c-1})\}$

$T \leftarrow T \cup \{u_{c-1,1}, u_{c-1,2}, \ldots, u_{c-1,\alpha}\}$

$T \leftarrow T - (B_c - \{v_c\})$

else

$T \leftarrow T - \{u_{c,\alpha}\}$

end

end

$M \leftarrow M \cup \{v_0\}$;

forall the $(u, v) \in S$ do

if $u \in M$ then

$M \leftarrow (M - \{u\}) \cup \{v\}$

end

end

return $M$

We conclude this section by presenting an example of Algorithm 4 in use in Figure A.2.
(1) A tree $T'$ with diametrical path $P = \{a, b, c, \ldots, m\}$

(2) Create shadow tree $T = S_{T', P}$ of $T'$ with no nested triangles. Add $m$ to $M$.

(3) Delete $\{k, l, m\}$ from $T$.

(4) Shift $B_c$ to $v_{c-1}$. Add $(g, f)$ to $S$. Add $j$ to $M$. 
(5) Delete \( \{h, i, j\} \) from \( T \).

(6) Swap \( Q_c \) and \( B_c \). \( P \) becomes \( P - B_c \cup Q_c \).

(7) Delete the nested triangle \( \Delta_f \).

(8) Shift \( B_c \) to \( v_{c-1} \). Add \((e, d)\) to \( S \). Add \( v \) to \( M \).
(9) Delete \( \{f, r, v\} \) from \( T \).

(10) Swap \( Q_c \) and \( B_c \).

(11) Shift \( B_c \). Add \((d, c)\) to \( S \). Add \( t \) to \( M \).

(12) Delete \( \{d, q, t\} \) from \( T \).

(13) Shift \( B_c \). Add \((b, a)\) to \( S \). Add \( e \) to \( M \).

(14) Delete \( \{b, c, e\} \) from \( T \). Add \( n \) to \( M \).

(15) \( M = \{m, j, v, t, e, n\}, S = \{(g, f), (e, d), (d, c), (b, a)\} \). Swap \( e \) and \( d \) in \( M \).

(16) A maximum multipacking \( M \) of \( T' \).

Figure 7: Example of Algorithm 4.
References

[1] J.R.S. Blair, P. Heggernes, S. Horton, F. Manne, Broadcast domination algorithms on interval graphs, series-parallel graphs, and trees. *Cong. Num.* **169** (2004), 55-77.

[2] I. Bouchemakh, M. Zemir, On the broadcast independence number of grid graph. *Graph Comb.* (2009), DOI:10.1007/s00373-012-1253-0.

[3] B. Brešar, S. Špacapan, Broadcast domination of products of graphs. *Ars. Comb.* **92** (2012), 303-320.

[4] R.C. Brewster, L. Duchesne, Broadcast domination and fractional multipackings. Manuscript, 2013.

[5] R.C. Brewster, G. McGillivray, F. Yang, Private communication, Nov. 28, 2013.

[6] R.C. Brewster, C.M. Mynhardt, L.E. Teshima, New bounds for the broadcast domination number of a graph. *Cent. Eur. J. Math.* **11**(7) (2013), 1334-1343.

[7] G. Chartrand, L. Lesniak, *Graphs and Digraphs*. Fourth Edition, Chapman & Hall, London, 2005.

[8] E. J. Cockayne, S. Herke, C. M. Mynhardt, Broadcasts and domination in trees. *Discrete Math.* **311** (2011), 1235–1246.

[9] G. Cornuéjols, *Combinatorial Optimization: Packing and Covering*. SIAM, Philadelphia, 2001.

[10] J. Dabney *A linear-time algorithm for broadcast domination in a tree*. Master’s Thesis, Clemson University, 2007.

[11] J. Dabney, B. C. Dean, S. T. Hedetniemi, A linear-time algorithm for broadcast domination in a tree. *Networks.* **53** (2009), 160–169.

[12] J. Dunbar, D. Erwin, T. Haynes, S. M. Hedetniemi, S. T. Hedetniemi, Broadcasts in graphs. *Discrete Appl. Math.* **154** (2006), 59–75.

[13] J. Dunbar, S. M. Hedetniemi, S. T. Hedetniemi, Broadcasts in trees. Manuscript, 2003.

[14] D. Erwin, *Cost domination in graphs*. Doctoral dissertation, Western Michigan University, 2001.

[15] D. Erwin, Dominating broadcasts in graphs. *Bull. Inst. Combin. Appl.* **42** (2004), 89–105.

[16] M. Farber, Characterizations of strongly chordal graphs. *Discrete Math.* **43** (1983), 173–189.

[17] M. Farber, Domination, independent domination, and duality in strongly chordal graphs. *Discrete Appl. Math.* **7** (1984), 115–130.
[18] D.R. Fulkerson, A.J. Hoffman, R. Oppenheim, On balanced matrices. *Math. Programming Stud.* 1 (1974), 120–132.

[19] B. Hartnell, C.M. Mynhardt, On the difference between broadcast and multipacking numbers of graphs. *Utilitas Math.*, to appear.

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

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

[22] P. Heggernes, D. Lokshtanov, Optimal broadcast domination in polynomial time. *Discrete Math.* 36 (2006), 3267–3280.

[23] S. Herke, *Dominating broadcasts in graphs*. Master’s thesis, University of Victoria, 2009. https://dspace.library.uvic.ca:8443/handle/1828/1479.

[24] S. Herke, C.M. Mynhardt, Radial Trees. *Discrete Math.* 309 (2009), 5950–5962.

[25] S. Lunney, *Trees with Equal Broadcast and Domination Numbers*. Master’s thesis, University of Victoria, 2011. https://dspace.library.uvic.ca:8443/handle/1828/3746.

[26] A. Meir and J. W. Moon, Relations between packing and covering numbers of a tree. *Pacific J. Math.* 61 (1975), 225–233.

[27] C.M. Mynhardt, L.E. Teshima, Broadcasts and multipackings in trees. To appear: *Util. Math.*

[28] D.J. Rose, Triangulated graphs and the elimination process. *J. Math. Anal. Appl.* 32 (1970), 597–609.

[29] S.M. Seager, Dominating broadcasts of caterpillars. *Ars Combin.* 88 (2008), 307–319.

[30] L.E. Teshima, *Broadcasts and multipackings in graphs*. Master’s thesis, University of Victoria, 2012. https://dspace.library.uvic.ca:8443/handle/1828/4341.