Closed Neighborhood Order Dominating Functions

Michael A. Henning*

Department of Mathematics, University of Natal
Pietermaritzburg, 3209 South Africa

Peter J. Slater

Mathematical Sciences
The University of Alabama in Huntsville
Huntsville, Alabama 35899, U.S.A.

Abstract

The linear programmimg formulations of graph domination and graph packing are duals. Likewise, fractional efficient domination for a graph G has as its dual the parameter discussed in this paper, namely the minimum weight of a real-valued function $f : V(G) \to [0, \infty)$ such that the sum of the weights in each closed neighborhood is at least its cardinality. The fractional and integer "closed neighborhood order domination" parameters, W_f and W, are introduced. Deciding if $W(G) \leq k$ is an NP-complete problem even for just the class of bipartite graphs or chordal graphs. A linear time algorithm for computing $W(T)$ for an arbitrary tree T is presented. Other theoretical and computational results are presented.

1 Introduction

Assume that the vertex set for each of the graphs G_1 and G_2 in Figure 1 on $n = 10$ and $n = 14$ vertices, respectively, represents a collection of towns, each of which supports its own standard sized fire station. In the event of a major fire in one town, fire fighters from an adjacent town can be expected to respond promptly enough to help. Note that four fire stations can respond to each of the towns u_2, u_3, u_4, u_7, u_8 and u_9 of G_1, but only two can respond to u_1, u_5, u_6 or u_{10}; and eight stations can respond to town v_7 or v_8 of G_2, but only two to each other v_i. Note that if we

*Research supported in part by the University of Natal and the South African Foundation for Research Development.

reduce the number of fire stations in G_1 from ten to eight and locate two of them at each of u_2, u_4, u_7, and u_9, then the same number of fire stations as before can still respond to each u_i for $1 \leq i \leq 10$. For G_2 we can reduce the number of fire stations from 14 to eight, and locate four of them at each of v_7 and v_8. We still have eight stations that can respond to each of v_7 and v_8. Further, the situation improves for each other v_i because now four stations can respond instead of only two.

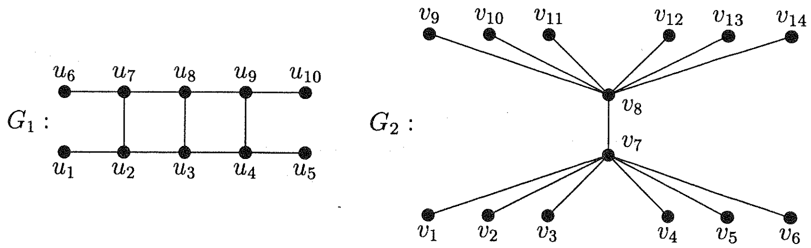


Figure 1: The graphs G_1 and G_2.

This motivates the following definition. Let G be a graph with vertex set V and edge set E. For a function $g : V \to [0, \infty)$ the *weight* of g is $w(g) = \sum_{v \in V} g(v)$, and for $S \subseteq V$ we define $g(S) = \sum_{v \in S} g(v)$ so $w(g) = g(V)$. We say a function $g : V \to \{0, 1, 2, 3, \ldots\}$ is a *closed neighborhood order dominating function*, or *CLOD-function*, if for each $v \in V$ we have $g(N[v]) \geq |N[v]|$, where $N[v]$ denotes the closed neighborhood of v, $N[v] = \{v\} \cup \{u \in V \,|\, uv \in E\}$. The *closed neighborhood order domination number* W is defined by $W(G) = min\,\{w(g) \,|\, g$ is a $CLOD$-function of $G\}$. The fractional version of this parameter is $W_f(G) = min\,\{w(g) \,|\, g : V \to [0, \infty)$ and $g(N[v]) \geq |N[v]|$ for each $v \in V\}$. That is, W and W_f are the minimum weights of nonnegative integer and real valued functions, respectively, which dominate in each closed neighborhood at least as well as the all ones assignment g_1 with $g_1(v) = 1$ for every $v \in V$. One can easily see that $W(G_1) = W(G_2) = 8$.

The parameters W and W_f along with closed neighborhood order packing parameters P and P_f were introduced in [10] where an Automorphism Class Theorem (ACT) for W_f and P_f is proven. (See Theorem 2 in this paper.) Packing parameters P and P_f are studied in [11].

We shall use standard graph theoretic terminology and notation. For example, for a graph $G = (V, E)$ with vertex set V and edge set E, the open neighborhood of $v \in V$ is $N(v) = \{u \in V \,|\, uv \in E\}$ and the closed neighborhood of v is $N[v] = \{v\} \cup N(v)$. Here, for $V = \{v_1, v_2, \ldots, v_n\}$ we let the closed neighborhood matrix be $N = [n_{i,j}]$ where $1 \leq i, j \leq n$, and $n_{i,j} = 1$ if $i = j$ or if $v_i v_j \in E$ and $n_{i,j} = 0$ otherwise.

Let $G = (V, E)$ be a graph. A packing in G is a set of vertices that are pairwise at distance at least 3 apart in G. That is, if S is a packing in G, then u and v in S implies $d(u, v) \geq 3$ where $d(u, v)$ denotes the distance between u and v. A dominating set $S \subseteq V$ for G is a vertex set with each $v \in V$ either in S or adjacent to a vertex of S. That is, S dominates if $\cup_{s\in S}N[s]=V$, and the domination number of G is $\gamma(G)$ which equals the minimum cardinality of a dominating set.

The efficient domination number of a graph, for which it is required that each vertex be dominated *at most once* and one seeks to dominate as many vertices as possible, is discussed in [1, 2, 3, 8, 9]. Because each vertex can be dominated at most once, the vertex set $S\subseteq V$ we use to dominate as many vertices as possible must be a packing. Define $F(G) = max\,\{|\cup_{s\in S} N[s]| \;:\; S \text{ is a packing}\}$. Note that if S is a packing then $|\cup_{s\in S} N[s]| = \sum_{s\in S}(1+deg\,s)$ because the closed neighborhoods of distinct vertices in S will be disjoint.

The fractional closed neighborhood order domination, W_f, can be derived as the linear programming dual of fractional efficient domination, F_f as shown in Figure 2. (In Figure 2, $\mathbf{d}$ denotes the column vector $(1+deg\,v_1, 1+deg\,v_2,\ldots,1+deg\,v_n)$.) The "multicoverage" application of the integer valued parameter W was presented earlier.

CLOD

$$W_f(G) = min\ \sum_{i=1}^{n} x_i$$

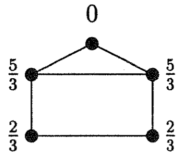
$$\text{subject to: } \begin{cases} N\cdot\mathbf{x} & \geq\ \mathbf{d} \\ x_i & \geq\ 0 \end{cases}$$

Efficient domination

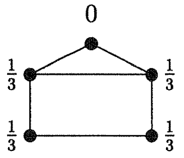
$$F_f(G) = max\ \sum_{i=1}^{n}(1+deg\,v_i)x_i$$

$$\text{subject to: } \begin{cases} N\cdot\mathbf{x} & \leq\ \mathbf{1} \\ x_i & \geq\ 0 \end{cases}$$

$W(G)=5$



$W_f(G) = \frac{14}{3} = F_f(G)$



$F(G)=4$

Figure 2: Linear programming formulation for W_f and F_f.

In the following theorem the equality follow by duality; two of the inequalities follow by definition; it is straightforward to see that $\gamma(G)\leq F(G)$ (if S be a maximal packing of G with $F(G)=\sum_{v\in S} deg\,v+1$, then $N[S]$ is a dominating set of G with $\gamma(G)\leq|N[S]|=F(G)$; and to see that $W(G)\leq n$ one simply considers the all ones function on V).

Theorem 1 [10]. *For any graph G of order n, we have $\gamma(G)\leq F(G)\leq F_f(G) = W_f(G)\leq W(G)\leq n$.*

Theorem 2 *(Automorphism Class Theorem)* [10]. *Given a graph $G=(V,E)$ and $g: V\to[0,\infty)$, let $g^*: V\to[0,\infty)$ be defined by $g^*(v)=\sum_{w\in[v]} g(w)/|[v]|$ where $[v]$ denotes the automorphism class of v. If g is a F_f, W_f or P_f-function, respectively, then so too is g^*.*

In Section 2 it is shown that deciding if $W(G)\leq k$ is an NP-complete problem for bipartite graphs and chordal graphs. Section 3 presents a linear-time algorithm for finding a minimum $CLOD$-function in an arbitrary tree. Section 4 presents some theoretical and computational results and Section 5 some concluding remarks.

2 NP-completeness

The following dominating set problem is well known to be NP-complete (see Garey and Johnson [7]), and remains NP-complete for the class of bipartite graphs, as shown by Dewdney [6], or chordal graphs, as shown by Booth [4] and Booth and Johnson [5]).

PROBLEM: Dominating set (**DOM**)

INSTANCE: A graph $H = (V, E)$ and a positive integer $k \leq |V|$.

QUESTION: Is $\gamma(G) \leq k$ (that is, is there a vertex set $S \subseteq V$ such that S is a dominating set with $|S| \leq k$)?

We will demonstrate a polynomial time reduction of the problem DOM to show that the following problem is also NP-complete.

PROBLEM: CLosed neighborhood Order Dominating (**CLOD**)

INSTANCE: A graph $G = (V, E)$ and a positive integer $\ell \leq |V|$.

QUESTION: Is $W(G) \leq \ell$ (that is, is there an integer valued function $g : V \to \{0, 1, 2, 3, \ldots\}$ such that $g(N[v]) \geq |N[v]|$ for each $v \in V$ and $w(g) \leq \ell$)?

Theorem 3 *Problem $CLOD$ is NP-complete, even when restricted to bipartite or chordal graphs.*

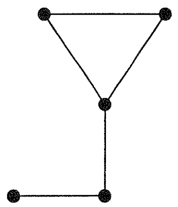
Proof. Any candidate solution g can be tested in polynomial time to see if $g(N[v]) \geq |N[v]| = 1 + deg\, v$ for each $v \in V$. Hence $CLOD \in NP$. To see that $CLOD$ is NP-complete it is next shown that a polynomial time algorithm for $CLOD$ could be used to solve DOM in polynomial time.

Starting with an instance $H = (V, E)$ and $k \leq |V| = n$ and $|E| = m$ for problem DOM, we can construct the graph G as follows. Starting with a copy of H where $V(H) = \{v_1, v_2, \ldots, v_n\}$, for each v_i we add $deg\, v_i$ copies of the path P_4. That is, we create a path $P_{i,j} = a_{i,j}, b_{i,j}, c_{i,j}, d_{i,j}$ for $1 \leq i \leq n$ and $1 \leq j \leq deg\, v_i$. Then we add all edges of the form $c_{i,j}v_i$. Note that for each $v_i \in V(H)$ the corresponding vertex in G has degree twice what it was in H. That is, $deg_G\, v_i = 2deg_H\, v_i$. (See Figure 3 for an example.) The number of paths we add to H is $\sum_{i=1}^{n} deg\, v_i = 2m$, so in forming G from H we have added $8m$ new vertices and $8m$ new edges. That is, $|V(G)| = n + 8m$ and $|E(G)| = 9m$. Graph G can be constructed from H in time polynomial in n. We note that if H is bipartite or chordal, then so too is G.

Let $\ell = 8m + k$. It suffices to show that $\gamma(H) \leq k$ if and only if $W(G) \leq \ell$. First, assume $\gamma(H) \leq k$, and let $S \subseteq V(H)$ be a dominating set of H with $|S| = k$. Define $g : V(G) \to \{0, 1, 2\}$ by $g(b_{i,j}) = g(c_{i,j}) = 2$ and $g(a_{i,j}) = g(d_{i,j}) = 0$ for $1 \leq i \leq n$ and $1 \leq j \leq deg\, v_i$, and let $g(v_i) = 1$ if $v_i \in S$ and $g(v_i) = 0$ if $v_i \notin S$. Then $g(N[b_{i,j}]) = 4 > |N[b_{i,j}]|$; $g(N[a_{i,j}]) = g(N[d_{i,j}]) = 2 = |N[a_{i,j}]| = |N[d_{i,j}]|$; $g(N[c_{i,j}]) = 5$ if $v_i \in S$ and is 4 if $v_i \notin S$, and $|N[c_{i,j}]| = 4$. Also, in G each v_i is adjacent to $deg_H\, v_i$ of the $c_{i,j}$'s, and v_i is adjacent to at least one vertex in S. Hence we have $g(N_G[v_i]) \geq 2deg_H\, v_i + 1 = deg_G\, v_i + 1 = |N_G[v_i]|$. Thus $W(G) \leq w(g) = 8m + |S| = 8m + k = \ell$.

Second, assume $W(G) \leq \ell$. Let $g : V(G) \to \{0, 1, 2, \ldots\}$ satisfy $g(N[v]) \geq |N[v]|$ for each $v \in V(G)$ (that is, g is a $CLOD$-function) and assume that $w(g) = W(G)$.

$H:$



$G:$

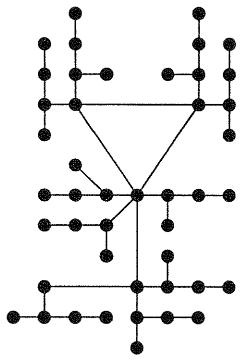


Figure 3: H is a $(n, m) = (5, 5)$ graph; G is a $(n + 8m, 9m) = (45, 45)$ graph.

If any $g(a_{i,j}) \geq 1$ one could increase $g(b_{i,j})$ by $g(a_{i,j})$ and set $g(a_{i,j}) = 0$, and the resulting function would be a $CLOD$-function of the same weight. Hence we can assume every $g(a_{i,j}) = 0$. Similarly, we can assume every $g(d_{i,j}) = 0$. Note that each $g(b_{i,j}) \geq 2 = |N[a_{i,j}]|$. If any $g(b_{i,j}) \geq 3$ one can modify g by increasing $g(c_{i,j})$ by $g(b_{i,j}) - 2$ and decreasing $g(b_{i,j})$ to 2. In fact, by "passing weights" from $a_{i,j}$ to $b_{i,j}$, from $d_{i,j}$ to $c_{i,j}$, from $b_{i,j}$ to $c_{i,j}$, and from $c_{i,j}$ to v_i as necessary we can assume that $g(a_{i,j}) = g(d_{i,j}) = 0$ and $g(b_{i,j}) = g(c_{i,j}) = 2$ for $1 \leq i \leq n$ and $1 \leq j \leq deg\, v_i$. Thus, $g(V(G) - V(H)) = 8m$, and so $g(V(H)) \leq \ell - 8m = k$. Furthermore, for every vertex v of H, the modified $CLOD$-function g satisfies $g(N_G[v] - N_H[v]) = deg_G v$. Since $g(N_G[v]) \geq deg_G v + 1$, we know therefore that $g(N_H[v]) \geq 1$ for every vertex v of H. Hence every vertex of H has positive weight under g or is adjacent to at least one vertex of H with positive weight under g. Thus, $S = \{v \in V(H) \mid g(v) > 0\}$ is a dominating set of H. Hence $\gamma(H) \leq |S| \leq g(V(H)) \leq k$.

So $\gamma(H) \leq k$ if and only if $W(G) \leq \ell = 8m + k$, and the proof is complete. □

3 A Linear Algorithm on Trees

Next we present a linear algorithm for finding a minimum closed neighborhood order dominating function (*CLOD-function*) f in a nontrivial tree T. The algorithm roots the tree T and associates various variables with the vertices of T as it proceeds. For any vertex v different from the root, the variable $Required(v)$ denotes the amount of domination still required by v. For any vertex v, the variable $MinValue(v)$ denotes the miminum value that may be assigned to v so that its children are (closed neighborhood order) dominated. The variable $ChildSum(v)$ denotes the sum of the values assigned by f to the children of v, while the variable $Sum(v)$ denotes the sum of the values assigned by f to v and the children of v.

Algorithm *CLOD* :

Input: *A nontrivial, rooted tree $T = (V, E)$ on n vertices with the vertices labeled from 1 to n so $label(w) > label(y)$ if the level of vertex w is less than the level of vertex y. [Note: the root of T is labeled n.]*

Output: *A minimum CLOD-function $f : V \to \{0, 1, 2, \ldots\}$.*

```
Begin
For $i \leftarrow 1$ to $n$ do
   1. If vertex $i$ is a leaf and $i < n$
         then
            $ChildSum(i) \leftarrow 0$ and $MinValue(i) \leftarrow 0$
         else
            $ChildSum(i) \leftarrow$ (sum of the values assigned by $f$ to the children of
                                       vertex $i$)
            $MinValue(i) \leftarrow$ (maximum of the values $Required(w)$ among all the
                                       children $w$ of vertex $i$).
   2. If $i < n$ then $f(i) \leftarrow MinValue(i)$
         else $f(i) \leftarrow max\,(MinValue(i), |N[i]| - ChildSum(i))$.
   3. $Sum(i) \leftarrow ChildSum(i) + f(i)$.
   4. If $i < n$ then $Required(i) \leftarrow max\,(0, |N[i]| - Sum(i))$.
End for
End CLOD
```

We now verify the validity of Algorithm *CLOD*.

Theorem 4 *Algorithm CLOD produces a minimum CLOD-function in a nontrivial tree.*

Proof. Let $T = (V, E)$ be a nontrivial tree of order n, and let f be the function produced by Algorithm *CLOD*. Then $f : V \to \{0, 1, 2, \ldots\}$.

Claim 1 *When Algorithm CLOD assigns a value $f(r')$ to the root r' of a subtree (or tree) T', the following two conditions will hold:*

1. *For any vertex $v \in T' - \{r'\}$, $f(N[v]) \geq |N[v]|$.*
2. *The value $f(r')$ assigned to r' is the minimum value it can receive given the values of its descendants under f.*

Proof. We proceed by induction on the order in which the vertices were labeled. The first vertex assigned a value will be a leaf. Vacuously, the first condition holds. In the case of a leaf i, $ChildSum(i) = 0$ and $MinValue(i) = 0$. Thus the leaf i will be assigned the value 0 in Step 2 of the algorithm and the second condition holds.

Next we assume that Algorithm $CLOD$ assigns values to the first k vertices so that Conditions 1 and 2 hold. We show that these conditions hold after the $(k+1)$st vertex is assigned a value.

We begin with Condition 1. Before the $(k+1)$st vertex is assigned a value, we can assume by the inductive hypothesis that all its descendants, other than its children, satisfy Condition 1. These descendants will continue to satisfy Condition 1 after the $(k+1)$st vertex is assigned a value. We show that any child w of vertex $k+1$ will also satisfy Condition 1. We note firstly that the value assigned to vertex $k+1$ in Step 2 is at least $MinValue(k+1)$, and in Step 1 of the algorithm $MinValue(k+1)$ is at least the value $Required(w)$ (≥ 0), so $f(k+1) \geq Required(w) \geq 0$. If w has $Sum(w) \geq |N[w]|$, then $f(N[w]) = f(k+1) + Sum(w) \geq Sum(w) \geq |N[w]|$. On the other hand, if $Sum(w) < |N[w]|$, then, in Step 4, $Required(w) = |N[w]| - Sum(w)$, so $f(N[w]) = f(k+1) + Sum(w) \geq Required(w) + Sum(w) = |N[w]|$. Thus, all descendants of the $(k+1)$st vertex will satisfy Condition 1.

Now consider Condition 2. The value assigned to r' in Step 2 is $MinValue(r')$ if the vertex r' is not the root of T and is at least $MinValue(r')$ if r' is the root of T. We show that for f to be a $CLOD$-function of T, the value for $f(r')$ must be at least $MinValue(r')$. We may assume that $MinValue(r') > 0$ for otherwise the result is immediate. Let w be a child of r' for which $MinValue(r') = Required(w)$ in the *else* statement in Step 1. Since $MinValue(r') > 0$, it follows that $Required(w) = |N[w]| - Sum(w)$ in Step 4. If r' was assigned a value less than $MinValue(r')$, then $f(N[w]) = f(r') + Sum(w) < MinValue(r') + Sum(w) = Required(w) + Sum(w) = |N[w]|$. It follows that for f to be a $CLOD$-function of T, the value for $f(r')$ must be at least $MinValue(r')$. Hence if r' is not the root of T, then the vertex r' satisfies Condition 2. If r' is the root of T, then in Step 2 the value assigned to r' is the larger of $MinValue(r')$ and $|N[r']| - ChildSum(r')$. If r' was assigned a value less than $|N[r']| - ChildSum(r')$, then $f(N[r']) = f(r') + ChildSum(r') < (|N[r']| - ChildSum(r')) + ChildSum(r') = |N[r']|$. It follows that for f to be a $CLOD$-function of T, the value for $f(r')$ must be at least $|N[r']| - ChildSum(r')$. Hence if r' is the root of T, then r' satisfies Condition 2. This completes the proof of the claim. □

Since $f(N[n]) = f(n) + ChildSum(n) \geq |N[n]|$, an immediate consequence of Claim 1 is that the function f produced by Algorithm $CLOD$ is a $CLOD$-function for T.

To show that the $CLOD$-function f obtained by Algorithm $CLOD$ is minimum, let g be any minimum $CLOD$-function for the rooted tree T. If $f \neq g$, then we will show that g can be transformed into a new minimum $CLOD$-function g' that will differ from f in fewer values than g did. This process will continue until $f = g$. Suppose, then, that $f \neq g$. Let v be the lowest labeled vertex for which $f(v) \neq g(v)$. Then *all* descendants of v are assigned the same value under g as under f. Hence it follows from Condition 2 of Claim 1 that $f(v) < g(v)$. Here the vertex v is not the root of T, for otherwise $f(V) < g(V) = W(T)$, which is impossible. Let w be the parent of v. Since all vertices at a lower level than v are assigned the same value under g as under f, Claim 1 implies that every vertex x at the same level as v has $f(x) \leq g(x)$. Let g' be the function produced by the following algorithm:

Algorithm 1 :

```
Begin
  1. Set $g'(x) = g(x)$ for all $x \in V$.
  2. Perform the following changes:
     while $(g'(v) > f(v))$ and $(g'(w) < f(w))$ do
         $INC(g'(w))$
         $DEC(g'(v))$
     end do
  3. Perform the following changes:
     while $(g'(v) > f(v))$ and $(g'(parent(w)) < f(parent(w)))$ do
         $INC(g'(parent(w)))$
         $DEC(g'(v))$
     end do
End
```

Claim 2 *The function g' produced by Algorithm* 1 *is a minimum $CLOD$-function of T that differs from f in fewer values than does g.*

Proof. The only vertices that have their closed neighborhood sums decremented are the children of v. However, these closed neighborhood sums under g' are at least as large as under f. Thus, since g and f are $CLOD$-functions, so too is g'. Furthermore, since g is minimum $CLOD$-function, so too is g'. It remains to show that g' differs from f in fewer values than does g.

The only possible vertices whose values under g and g' differ are v, w, and $parent(w)$. By Algorithm 1, if $g(w) = f(w)$, then $g'(w) = f(w)$, and if $g(parent(w)) = f(parent(w))$, then $g'(parent(w)) = f(parent(w))$. We show that $g'(v) = f(v)$. If this is not the case, then $g'(v) > f(v)$. It follows then from Algorithm 1 that $g'(w) \geq f(w)$ and $g'(parent(w)) \geq f(parent(w))$. By the ordering scheme, the value assigned to each child of w under g, and therefore under g', is greater than or equal to its value under f. Thus, since $g'(v) > f(v)$, the sum of the values assigned to the children of w under g' is greater than under f. Consequently, $g'(N[w]) \geq f(N[w])+1$. So the vertex v and all its neighbors, including w, will have smaller neighborhood sums under f than under g'. Hence, the function obtained from g' by reassigning to the vertex v the value $g'(v) - 1$, and leaving the values of all other vertices unchanged, is a $CLOD$-function. This, however, contradicts the minimality of g'. Hence $g'(v) = f(v)$. This completes the proof of Claim 2 and of Theorem 4. □

4 Examples and bounds

In Figure 1 we have two examples where $W(G) < |V(G)| = n$. We first show that $W(G)$ and n can differ by an arbitrary amount. In fact, $W(G)/n$ can be arbitrarily small. Consider the tree $T_{j,k}$ in Figure 4 created as follows. A vertex u is given j neighbors, $N(u) = \{v_1, v_2, \ldots, v_j\}$. Each v_i is made adjacent to k end-vertices. Thus $|V(T_{j,k})| = jk + j + 1$. The function $g : V(T_{j,k}) \to \{0, 2, k\}$ with $f(u) = k$, $f(v_i) = 2$ for $1 \leq i \leq j$, and $f(x) = 0$ for each end-vertex x is a $CLOD$-function with $w(g) = W(T_{j,k}) = 2j + k$. (Note that $F(T_{j,k}) = 2j + k$ also.) For a fixed k, letting j get large makes $W(T_{j,k})/|V(T_{j,k})| = (2j + k)/(jk + j + 1)$ approach $2/(k + 1)$. Thus we have the next proposition.

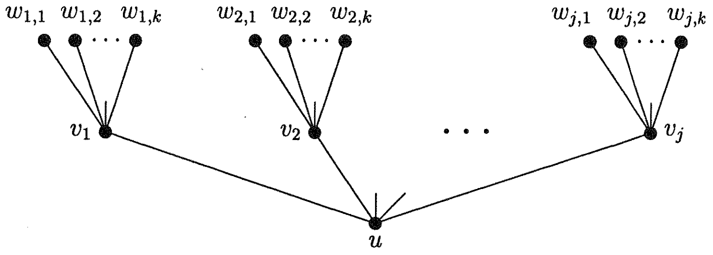

$w_{1,1}$ $w_{1,2}$ $w_{1,k}$
$w_{2,1}$ $w_{2,2}$ $w_{2,k}$
$w_{j,1}$ $w_{j,2}$ $w_{j,k}$
v_1
v_2
v_j
u

Figure 4: The tree $T_{j,k}$.

Proposition 1 *The ratio $W(G)/|V(G)|$ can be made arbitrarily small, even for trees.*

For the cycle C_n with $n = 3k + r$ and $0 \leq r \leq 2$ we have $F(C_n) = 3k$, but by letting $g(v) = 1/3$ for every $v \in V(C_n)$ we get $F_f(C_n) = n$, and so $W(C_n) = n$. More generally, if G is regular of degree r define $g : V(G) \to [0, \infty)$ by $g(v) = 1/(r + 1)$ for every $v \in V(G)$, and we see that $F_f(G) = n$. Thus we have the following result.

Proposition 2 *For every r-regular graph of order n, $n = F_f(G) = W_f(G) = W(G)$.*

Let G be a graph of order n and size m with vertex set $V(G) = \{v_1, v_2, \ldots, v_n\}$. Intuitively, when G is not regular it seems that it would be better to place higher weights on the vertices of large degree in order to better reach the amount of domination required. Specifically, each v_i must be dominated by a weight of $1 + deg\, v_i$ in $N[v_i]$, so the total amount of domination required is $\sum_{i=1}^{n}(1 + deg\, v_i) = n + 2m$. However, for the graph G shown in Figure 5, $W(G) = 29$ and the unique $CLOD$-function g with $w(g) = 29$ is illustrated. We note that while $g(v) = 5$ we have $g(u) = g(w) = 0$ for the vertices u and w of maximum degree $\Delta(G) = 4$. We do get the following degree sequence bound for $W(G)$.

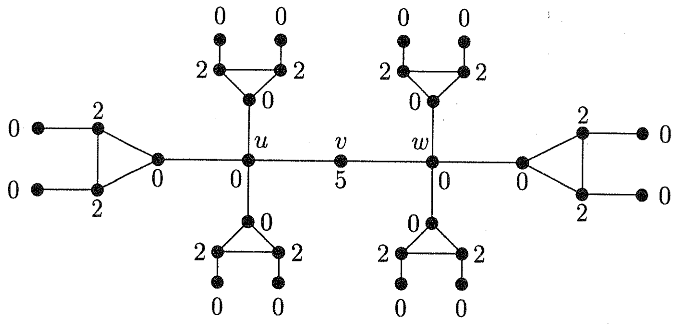


Figure 5: The graph G.

Theorem 5 *Let $G = (V, E)$ be a graph of order n and size m, the degrees d_i of whose vertices v_i satisfy $d_1 \geq d_2 \geq \cdots \geq d_n$. If t is the largest integer for which*

$$(d_1+1)^2 + (d_2+1)^2 + \cdots + (d_t+1)^2 + k(d_{t+1}+1) \leq n + 2m,$$

for some k with $0 \leq k \leq d_{t+1}$, then $W(G) \geq W_f(G) \geq d_1 + d_2 + \ldots d_t + t + k$.

Proof. Let $g : V \to [0, \infty)$ be a CLOD-function satisfying $w(g) = W(G)$. We consider the total amount of domination done by g, namely the sum $N = \sum\sum g(u)$, where the outer sum is over all $v \in V$ and the inner sum is over all $u \in N[v]$. Since $\sum_{u \in N[v]} g(u) = g(N[v]) \geq deg\, v + 1$ for each $v \in V$,

$$N \geq \sum_{v \in V} (deg\, v + 1) = n + 2m. \tag{1}$$

The sum N counts the value $g(u)$ exactly $deg\, u + 1$ times for each $u \in V$, so

$$N = \sum_{u \in V} (deg\, u + 1) g(u) = \sum_{i=1}^{t} (d_i + 1) g(v_i) + \sum_{i=t+1}^{n} (d_i + 1) g(v_i). \tag{2}$$

Suppose $g(v_i) \leq d_i + 1$ for all i and assume that $W(G) < d_1 + d_2 + \ldots d_t + t + k$. Then

$$\sum_{i=1}^{t} (d_i + 1) + k > W(G) = w(g) = \sum_{i=1}^{t} g(v_i) + \sum_{i=t+1}^{n} g(v_i). \tag{3}$$

Since each $g(v_i) \leq d_i + 1$, and $d_1 \geq d_2 \geq \cdots \geq d_n$, the sum in (2) is a maximum when $g(v_i) = d_i + 1$ for $1 \leq i \leq t$, i.e., when $\sum_{i=1}^{t} g(v_i) = \sum_{i=1}^{t} (d_i + 1)$. This would imply, by (3), that $\sum_{i=t+1}^{n} g(v_i) < k$. Thus, by (2),

$$\begin{aligned} N &\leq \textstyle\sum_{i=1}^{t} (d_i+1)^2 + (d_{t+1}+1) \sum_{i=t+1}^{n} g(v_i) \\ &< \textstyle\sum_{i=1}^{t} (d_i+1)^2 + (d_{t+1}+1)k \\ &\leq n + 2m, \qquad \text{(by assumption)} \end{aligned}$$

which contradicts equation (1). Hence if each $g(v_i) \leq d_i + 1$ then the result follows.

Because $w(g) = W(G)$, if $g(v) > 0$ and one decreases $g(v)$ then we no longer have a CLOD-function. That is, there must be a vertex $u \in N[v]$ such that $g(N[u]) = deg\, u + 1$. Observe that if $g(v) > deg\, v + 1$ then we must have $deg\, u > deg\, v$.

Assume that we have vertices v_i with $g(v_i) > d_i + 1$. For each such v_i select one $u_i \in N[v_i]$ such that $g(N[u_i]) = deg\, u_i + 1$. As noted, $deg\, u_i > deg\, v_i$, so u_i corresponds to a v_j where $j < i$. Also, we might have some $u_i = u_h$ with $h \neq i$. Let $g^* : V \rightarrow [0, \infty)$ be the function with $w(g^*) = w(g)$ obtained from g as follows. For each v_i with $g(v_i) > d_i + 1$ let $g^*(v_i) = 0$ and increase the function value at u_i by $g(v_i)$. We have $w(g) = w(g^*)$ and each $g^*(v_i) \leq d_i + 1$, and $\sum_{i=1}^{n}(d_i + 1)g^*(v_i) > \sum_{i=1}^{n}(d_i + 1)g(v_i)$. Finally, if $w(g^*) = w(g) = W(G) < d_1 + d_2 + \ldots d_t + t + k$, then $N = \sum_{i=1}^{n}(d_i + 1)g(v_i) < \sum_{i=1}^{n}(d_i + 1)g^*(v_i) < n + 2m$, a contradiction, completing the proof. □

To illustrate Theorem 5, consider the graph G_1 of order $n = 10$ and size $m = 11$ shown in Figure 1. The graph G_1 has degree sequence $d_1, d_2, \ldots, d_{10}$, where $d_i = 3$ for $1 \leq i \leq 6$ and $d_i = 1$ for $7 \leq i \leq 10$. The largest integer t for which $\sum_{i=1}^{t}(d_i + 1)^2 + k(d_{t+1} + 1) \leq n + 2m = 32$, where $0 \leq k \leq d_{t+1}$, is $t = 2$ with $k = 0$. Hence applying Theorem 5, we get $W(G) \geq d_1 + d_2 + 2 + 0 = 8$. As observed earlier, $W(G_1) \leq 8$, whence $W(G_1) = 8$.

We conclude this section by considering the complete multipartite graph $G \cong K_{n_1, n_2, \ldots, n_k}$ with $n = n_1 + n_2 + \cdots + n_k$ vertices, independent vertex sets $S_1, S_2, \ldots, S_k$ with $|S_i| = n_i$, and all edges uv with $u \in S_i$, $v \in S_j$ and $i \neq j$.

Proposition 3 $W_f(K_{n_1, n_2, \ldots, n_k}) = W(K_{n_1, n_2, \ldots, n_k}) = n$.

Proof. If any vertex v in a graph G has $deg\, v = n - 1$ then $W_f(G) \geq |N[v]| = n$, so we can assume $n_i \geq 2$ for $1 \leq i \leq k$. Let $g : V(K_{n_1, n_2, \ldots, n_k}) \rightarrow [0, \infty)$ be a *CLOD-function* with $w(g) = W_f(K_{n_1, n_2, \ldots, n_k})$. By the Automorphism Class Theorem we can assume that $u, v \in S_i$ implies $g(u) = g(v) = a_i$, say, for $1 \leq i \leq k$.

If all $a_i = 1$, then $w(g) = n$. Suppose some $a_i > 1$, and let $v \in S_i$. We show that $w(g) \geq n$. We have $\sum_{x \in N[v]} g(x) \geq |N[v]| = n - n_i + 1$. Therefore $g(V - S_i) \geq n - n_i + 1 - a_i$, and $g(V) = w(g) \geq n - n_i + 1 - a_i + n_i \cdot a_i$. If $w(g) < n$, then $-n_i + 1 - a_i + n_i \cdot a_i < 0$ implies $n_i(a_i - 1) < a_i - 1$ and $n_i < 1$, a contradiction. Finally, suppose some $a_i < 1$ and every $a_j \leq 1$. Then for $v \in S_i$ we have $\sum_{x \in N[v]} g(x) \leq a_i + n - n_i < n - n_i + 1 = |N[v]|$, a contradiction. □

5 Open problems

In the course of this investigation we encountered a number of problems which we have yet to settle. A partial listing of these problems follows.

1. A graph $G = (V, E)$ is *efficiently dominatable* if there is a dominating set $S \subseteq V$ such that for each $v \in V$ we have $|N[v] \cap S| = 1$. Is it true that a tree T of order n is efficiently dominatable if and only if $W(T) = n$?

2. More generally, is it true that $F(T) = W(T)$ for all trees T?

3. Is it true that, given a graph G of order n, the problem of deciding whether $W(G) < n$ is NP-complete? (Note that in [2] it is shown that deciding if $F(G) = n$ is NP-complete.)

4. Given a positive integer n, determine the minimum of the ratios $\frac{W(G)}{n}$ among all graphs G of order n.

References

[1] D.W. Bange, A.E. Barkauskas, and P.J. Slater, Disjoint dominating sets in trees, Sandia Laboratories Report, SAND78-1087J, 1978.

[2] D.W. Bange, A.E. Barkauskas, and P.J. Slater, Efficient dominating sets in graphs, *Applications of Discrete Math.*, SIAM, Philadelphia, 1988, 189-199.

[3] D.W. Bange, A.E. Barkauskas, L. Host, and P.J. Slater, Efficient near-domination of grid graphs, *Congressus Numerantium* **58** (1987), 83-92.

[4] K.S. Booth, Dominating sets in chordal graphs, Research Report CS-80-34, University of Waterloo, 1980.

[5] K.S. Booth and J.H. Johnson, Dominating sets in chordal graphs, *SIAM J. Comput.* **11**(1982), 191-199.

[6] A.K. Dewdney, Fast Turing reductions between problems in NP, Report 71, University of Western Ontario, 1981.

[7] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York (1979).

[8] D.L. Grinstead and P.J. Slater, Fractional domination and fractional packing in graphs, *Congressus Numerantium* **71** (1990), 153-172.

[9] D.L. Grinstead and P.J. Slater, A recurrence template for several domination related parameters in series-parallel graphs, *Discrete Applied Math.* **54** (1994), 151–168.

[10] P.J. Slater, Closed neighborhood order domination and packing, *Congressus Numerantium* **97** (1993), 33-43.

[11] P.J. Slater, Packing into closed neighborhoods, *Bull. Inst. Combin. Applic.* **13** (1995), 23–33.

(Received 30/10/96)