Discrete Mathematics 86 (1990) 239–254
North-Holland

ON MINIMUM DOMINATING SETS WITH MINIMUM INTERSECTION

Dana L. GRINSTEAD* and Peter J. SLATER* **
Department of Mathematics and Statistics, The University of Alabama in Huntsville, Huntsville, AL 35899, USA

Received 2 December 1988

In the developing theory of polynomial/linear algorithms for various problems on certain classes of graphs, most problems considered have involved either finding a single vertex set with a specified property (such as being a minimum dominating set) or finding a partition of the vertex set into such sets (for example, a partition into the maximum possible number of dominating sets). Alternatively, one might be interested in the cardinality of the set or the partition. In this paper we introduce an intermediate type of problem. Specifically, we ask for two minimum dominating sets with minimum intersection. We present a linear algorithm for finding two minimum dominating sets with minimum possible intersection in a tree T, and we show that simply determining whether or not there exist two disjoint minimum dominating sets is NP-hard for arbitrary bipartite graphs.

1. Introduction

Given a graph $G=(V, E)$, a vertex subset $S \subseteq V$ is independent if no two vertices in S are adjacent; $\beta(G)$ will here denote the maximum number of vertices in an independent set; G is k-colorable if $V=V_1 \cup V_2 \cup \cdots \cup V_k$ where each V_i is independent; and the chromatic number $\chi(G)$ is the minimum k such that G is k-colorable. Vertex subset $D \subseteq V$ is a dominating set if each $v \in V-D$ is adjacent to at least one vertex in D; $\gamma(G)$ here denotes the minimum number of vertices in a dominating set; G is k-domatic if V can be partitioned into k sets $V_1, V_2, \ldots, V_k$ such that each V_i is a dominating set for G; and the domatic number of G is the maximum k such that G is k-domatic. Determining if $\beta(G) \geqslant K$ is an NP-complete problem even for cubic planar graphs (Garey, Johnson and Stockmeyer [17]); deciding if G is K-colorable is NP-complete even for planar graphs of maximum degree four (Karp [23]); deciding if $\gamma(G) \leqslant K$ is NP-complete for planar graphs of maximum degree three (Garey and Johnson [19]) and for bipartite graphs (Dewdney [16]); and for the domatic number problem introduced by Cockayne and Hedetniemi [13] determining if the domatic number of G is at least K is NP-complete (Garey, Johnson and Tarjan [18]).

Much of the extensive amount of research in graph algorithms has been concerned with developing polynomial time algorithms for NP-complete problems restricted to appropriate classes of graphs. Indeed, many linear time algorithms

* Research supported in part by the U.S. Office of Naval Research Grant N00014-86-K-0745.
** Dr. Slater thanks those at Clemson University for their hospitality during the Fall of 1987 and gratefully acknowledges support under the U.S. Office of Naval Research Grant N00014-86-K-0693.

have been developed. As examples, we have linear algorithms for minimum domination in trees (Cockayne, Goodman and Hedetniemi [12]), R-domination in trees (Slater [32]) and block graphs (Chang and Nemhauser [14]), independent domination in trees (Beyer, Proskurowski, Hedetniemi and Mitchell [8]), independent domination and total domination in seriesparallel graphs (Hedetniemi, Laskar and Pfaff [28]), domination in seriesparallel graphs (Kikuno, Yoshida and Kakuda [24]), locating-dominating sets in seriesparallel graphs (Colbourn, Slater and Stewart [15]), and dominating subforests of a tree (Lawler and Slater [26]). Many other domination related algorithmic papers have appeared, as have many related to finding independent sets in graphs. In general, much work has been done to develop polynomial/linear algorithms for finding a (minimum/maximum) vertex or edge set S with a specified property. Further, problems involving partitions of vertex set V have been investigated. For example, as reported in Johnson [22], Bodlaender [9] has developed a k-chromatic number algorithm for partial h-trees that is polynomial for fixed k and h.

In fact, a general theory of linear algorithms is being developed. Especially notable is the thesis of Wimer [35], with other notable papers including Takamizawa, Nishizeki and Saito [33], Bern, Lawler and Wong [6], Arnborg and Proskurowski [1], and the work of Robertson and Seymour, including [29].

In this paper we introduce an intermediate type of problem. The general type of problem is defined by asking for more than one vertex set with required properties, but not necessarily for a partition of V. A general treatment of such problems is contained in Grinstead [20]. Some previous work on finding a pair of disjoint dominating sets having some property P appears in Bange, Barkauser and Slater [2–5]. Here we relax the requirement of disjointness and ask for two minimum dominating sets with minimum possible intersection. We let $M_\gamma(G)$ denote the minimum cardinality of the intersection of two minimum dominating sets in G. Note that if G has a unique minimum dominating set D, then $M_\gamma(G)=\gamma(G)=|D|$. In the next section, we show that simply determining if there exist two disjoint minimum dominating sets is NP-hard for arbitrary bipartite graphs. Section 3 contains a linear algorithm for computing $M_\gamma(T)$ for a tree T. The algorithm works by a single pass over the endpoint list of T (described in Section 3). Then in Section 4, we note that two such sets can actually be obtained by an additional backward pass through the endpoint list, and briefly discuss how the procedure can be extended to cover series-parallel graphs.

2. Determining $M_\gamma(G)$ is NP-hard

Having defined $M_\gamma(G)$ to be the minimum cardinality of the intersection of two minimum dominating sets in G, we can pose the following decision problem. Given a graph G and a nonnegative integer K, is $M_\gamma(G)\leq K$? In this section we show that simply determining whether or not $M_\gamma(G)=0$ is NP-hard for bipartite graphs G. As was pointed out to us by a referee, our DISJOINT MINIMUM DOMINATING SETS problem is in NP^{NP}, the class of languages recognizable nondeterministically in polynomial time with the aid of an oracle from NP. Given an oracle to test if $\gamma(G)=k$, such a nondeterministic algorithm is as follows: Guess at two sets $D1$ and $D2$ and verify that they are disjoint dominating sets. Using the oracle, verify that each has cardinality $\gamma(G)$.

We next describe a polynomial time reduction from NOT-ALL-EQUAL 3SAT (see Schaefer [30]) to the problem of determining if $M_\gamma(G)=0$ for bipartite G, which implies that this DISJOINT MINIMUM DOMINATING SETS problem is NP-hard. The NOT-ALL-EQUAL 3SAT problem appears in Garey and Johnson [19, p. 259], and [19] contains a complete discussion of the theory of NP-completeness.

NOT-ALL-EQUAL 3SAT

Instance: Set U of variables, collection C of clauses over U such that each clause $c \in C$ has $|c|=3$.
Question: Is there a truth assignment for U such that each clause in C has at least one true literal and at least one false literal?

DISJOINT MINIMUM DOMINATING SETS

Instance: Graph G.
Question: Does G have two disjoint minimum dominating sets?

Let $U=\{u_1, u_2, \ldots, u_n\}$. Given $C=c_1 \wedge c_2 \wedge \cdots \wedge c_m$ where $c_i=(s_{i1} \vee s_{i2} \vee s_{i3})$ and each s_{ij} is u_h or $\bar{u}_h$ for some $1 \leqslant h \leqslant n$, we show here how to construct a graph G (in time polynomial in m) such that U has a NOT-ALL-EQUAL 3SAT truth assignment for C if and only if G has DISJOINT MINIMUM DOMINATING SETS. Hence a polynomial time algorithm for the latter decision problem would imply a polynomial time algorithm for the former known NP-complete problem. The graph G will contain $3m$ copies of the graph H in Fig. 1. We need to note that H is bipartite with vertices labelled u_{ij} and $\bar{u}_{ij}$ in the same set of the bipartition, and the only vertices in H adjacent to other vertices of G will be u_{ij} and $\bar{u}_{ij}$ (so that the degrees in G satisfy $\deg_G(v)=\deg_G(w)=7$ and $\deg_G(x_i)=2$ for $1 \leqslant i \leqslant 14$). Each copy of H in G will be called an H-subgraph with designated vertices u_{ij} and $\bar{u}_{ij}$. Letting D be any minimum dominating set for G, the following observations are easy to verify. Set D must contain at least one of u_{ij} and $\bar{u}_{ij}$ (consider x_1, x_2, x_3); $|D \cap V(H)| \geqslant 3$; and if $|D \cap V(H)|=3$ then $D \cap V(H)$ is $\{v, w, u_{ij}\}$, $\{v, w, \bar{u}_{ij}\}$, $\{u_{ij}, w, x_{14}\}$ or $\{\bar{u}_{ij}, v, x_{13}\}$. In particular, if G has two disjoint minimum dominating sets, then one contains $\{u_{ij}, w, x_{14}\}$ and the other contains $\{\bar{u}_{ij}, v, x_{13}\}$.

For each clause $c_i=(s_{i1} \vee s_{i2} \vee s_{i3})$ we construct a graph G_i on 58 vertices like the one illustrated in Fig. 2 as follows. Suppose $s_{i1}=u_a$ or $\bar{u}_a$, $s_{i2}=u_b$ or $\bar{u}_b$, and $s_{i3}=u_c$ or $\bar{u}_c$. Let G_i contain three copies of H with designated vertices u_{ia} and $\bar{u}_{ia}$, u_{ib} and $\bar{u}_{ib}$, and u_{ic} and $\bar{u}_{ic}$. Each of c_{1i} and c_{2i} is connected to u_{ia} if $s_{i1}=u_a$, to $\bar{u}_{ia}$ if $s_{i1}=\bar{u}_a$, to u_{ib} if $s_{i2}=u_b$, to $\bar{u}_{ib}$ if $s_{i2}=\bar{u}_b$, to u_{ic} if $s_{i3}=u_c$, and to $\bar{u}_{ic}$ if

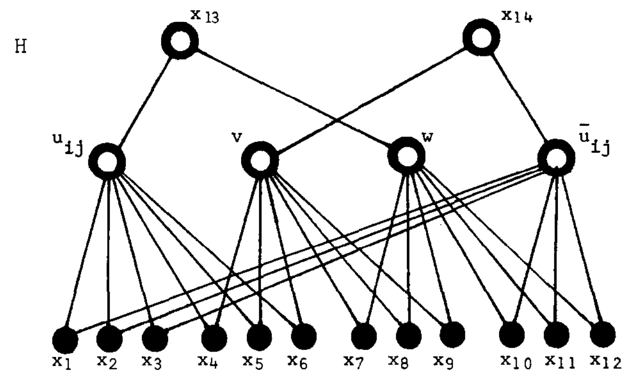

Fig. 1. A 'building block' H for graph G.

$s_{i3} = \bar{u}_c$. Each of c_{3i} and c_{4i} is connected to the three designated vertices not adjacent to c_{1i} and c_{2i}.

Let G be the graph containing disjoint copies of $G_1, G_2, \ldots, G_m$ to which we add the following vertices and edges. For each occurrence of a u_h or $\bar{u}_h$ with $1 \leqslant h \leqslant n$ in distinct clauses c_i and c_j add four vertices of degree two as follows. Assume s_{ir} is u_h or $\bar{u}_h$ and assume s_{jt} is u_h or $\bar{u}_h$ where $1 \leqslant i < j \leqslant m$ and $1 \leqslant r, t \leqslant 3$. Let two of the four vertices be adjacent to $\bar{u}_{ih}$ and to u_{jh}, and let the other two be adjacent to u_{ih} and to $\bar{u}_{jh}$. The graph G is illustrated in Fig. 3 for $C = (u_1 \vee u_2 \vee u_3) \wedge (\bar{u}_1 \vee \bar{u}_2 \vee u_4) \wedge (u_2 \vee u_3 \vee u_5)$.

Note that G contains $58m$ vertices in $\bigcup_{i=1}^{m} G_i$. Further, each u_{ij} or $\bar{u}_{ij}$ is adjacent to at most $2(m-1)$ vertices in $G - \bigcup_{i=1}^{m} G_i$, and so there are at most $6m^2 + 52m$ vertices in G, and G can be constructed from C in time polynomial in m.

Theorem. *Graph G has two* DISJOINT MINIMUM DOMINATING SETS *(that is, $M_\gamma(G) = 0$) if and only if U has a* NOT-ALL-EQUAL 3SAT *truth assignment for C.*

Proof. First, assume there is a NOT-ALL-EQUAL 3SAT truth assignment for

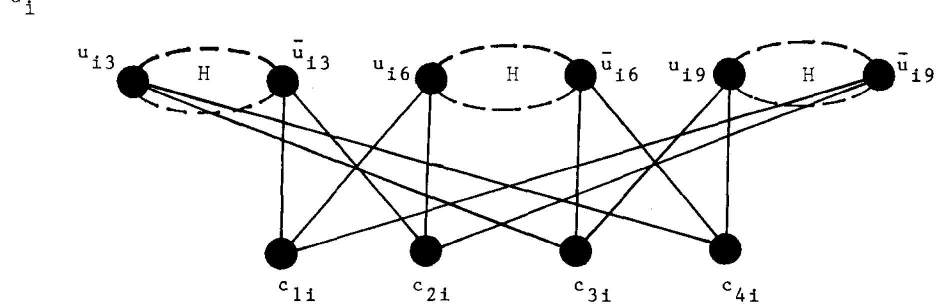

Fig. 2. A larger 'building block' G_i for G for $C_i = (\bar{u}_3 \vee u_6 \vee \bar{u}_9)$.

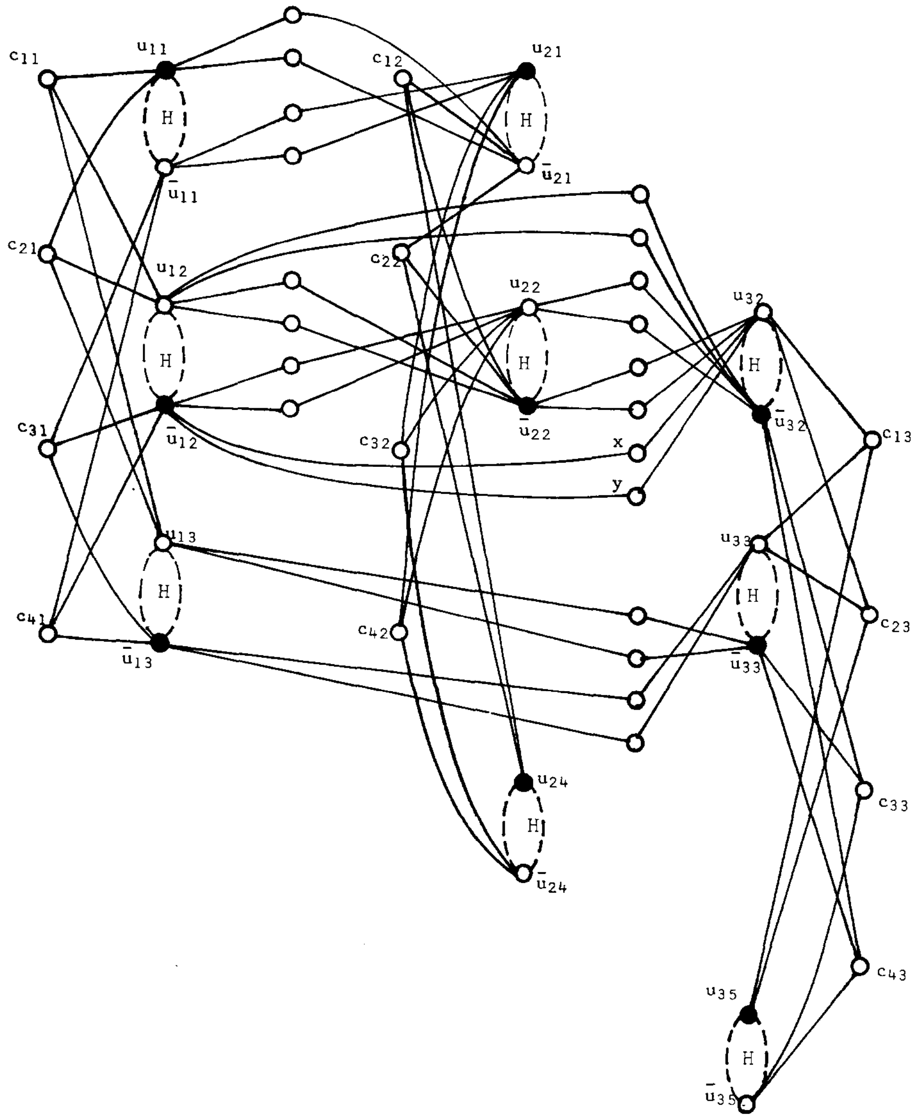

Fig. 3. Graph G for $C=(u_1 \vee u_2 \vee u_3) \wedge (\bar{u}_1 \vee \bar{u}_2 \vee \bar{u}_4) \wedge (u_2 \vee u_3 \vee u_5)$.

C. Let $S1$ consist of those u_i in U that receive the value true and $S2=U-S1$. Construct two vertex subsets $D1$ and $D2$ of $V(G)$ as follows. For each u_{ij} in G if $u_j \in S1$ then place u_{ij} and the corresponding w and x_{14} of its building block H (as in Fig. 1) in $D1$ and place $\bar{u}_{ij}$ and the corresponding v and x_{13} in $D2$, and if $u_j \in S2$ then place u_{ij}, w and x_{14} in $D2$ and $\bar{u}_{ij}$, v and x_{13} in $D1$. For example, if $C=(u_1 \vee u_2 \vee u_3) \wedge (\bar{u}_1 \vee \bar{u}_2 \vee \bar{u}_4) \wedge (u_2 \vee u_3 \vee u_5)$ then one NOT-ALL-EQUAL 3SAT truth assignment is to let $S1=\{u_1, u_4, u_5\}$, and the nine darkened u_{ij} and

$\bar{u}_{ij}$ in Fig. 3 are placed in $D1$ (with two additional vertices from each H) with the nine undarkened u_{ij} and $\bar{u}_{ij}$ going into $D2$.

As previously noted, every minimum dominating set for G must contain at least three or more vertices from each H. Thus $\gamma(G) \geq 9m$. Clearly $D1 \cap D2 = \emptyset$ and $|D1| = |D2| = 9m$, and it is straighforward to see that each of $D1$ and $D2$ dominates G. Hence, G has two DISJOINT MINIMUM DOMINATING SETS.

Conversely, assume G has two DISJOINT MINIMUM DOMINATING SETS, say $D1$ and $D2$. As noted, for each u_{ij} either $D1$ or $D2$ contains u_{ij} and two specified vertices from its H-subgraph, and the other contains $\bar{u}_{ij}$ and two specified vertices from this same H subgraph. Suppose u_{ik} and u_{jk} are vertices of G with $i \neq j$ (for example, u_{12} and u_{32} in Fig. 3). To see that u_{ik} and $\bar{u}_{jk}$ cannot both be in $D1$ or both be in $D2$, note that if u_{ik} and $\bar{u}_{jk}$ are in $D1$ (and so $\bar{u}_{ik}$ and u_{jk} are in $D2$), then $D1$ must also contain the two vertices x and y of degree two adjacent to $\bar{u}_{ik}$ and u_{jk} (for example, vertices x and y in Fig. 3). Recognizing that one of the two specified vertices in $D1$ from the H-subgraph of $\bar{u}_{jk}$ dominates u_{jk}, the set $D1 - x - y + \bar{u}_{ik}$ would also be a dominating set, contradicting the minimality of $D1$. Consequently, if u_{ik} and u_{jk} are vertices of G with $i \neq j$ then both are in $D1$ or both are in $D2$. Therefore, the following is a well defined truth assignment for U. For each $u_k \in U$ find a u_{ik} in G, and let u_k be true if $u_{ik} \in D1$ and false if $u_{ik} \in D2$.

It remains only to show that each clause c_i has at least one true (respectively, false) literal. If not, we can assume clause c_i has three true literals. Then each of c_{3i} and c_{4i} is adjacent to three vertices in $D2$, so c_{3i} and c_{4i} are in $D1$. Letting x be one of the vertices adjacent to c_{3i} and c_{4i}, we see that $D1 - c_{3i} - c_{4i} + x$ is a dominating set, contradicting the minimality of $D1$. Using Fig. 3 as an example, if u_1 and u_2 and u_3 are true then $\{\bar{u}_1, \bar{u}_2, \bar{u}_3\} \subset D2$ and $\{u_{11}, u_{12}, u_{13}, c_{31}, c_{41}\} \subset D1$, and $D1 - c_{31} - c_{41} + \bar{u}_{13}$ would be a dominating set strictly smaller than dominating set $D1$. It follows that U has a NOT-ALL-EQUAL 3SAT truth assignment for C. □

3. A linear algorithm for determining $M\gamma(T)$

In this section we will present a linear algorithm for determining $M_\gamma(T)$, the minimum cardinality of the intersection of two minimum dominating sets of tree T. Section 4 will note how an algorithm for finding two minimum dominating sets whose intersection has cardinality $M_\gamma(T)$ can be derived.

Without loss of generality, it will be assumed that all trees are rooted at some vertex which can be chosen arbitrarily. This will enable us to use recursive representations of trees. Given a rooted tree T, we will represent T by the number of nodes in T, say n, an endnode list $\mathrm{EL} = (v_1, v_2, \ldots, v_n)$, and an associated parent list $\mathrm{PA} = (u_1, u_2, \ldots, u_{n-1})$. The endnode list is any enumeration of the nodes of T in which each node precedes its parent. In the associated

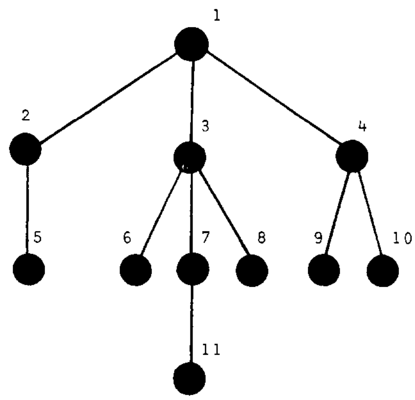

Fig. 4. A tree with EL = (5, 6, 11, 7, 8, 9, 10, 2, 3, 4, 1) and PA = (2, 3, 7, 3, 3, 4, 4, 1, 1, 1).

parent list, each u_i is the parent of v_i in T. Note that PA has length $n - 1$ and not n, since the root v_n has no parent. For example, the tree of Fig. 4 may be represented by $n = 11$, EL = (5, 6, 11, 7, 8, 9, 10, 2, 3, 4, 1), and PA = (2, 3, 7, 3, 3, 4, 4, 1, 1, 1).

These lists can be constructed for a tree of n nodes in time O(n) (see e.g. [25, 34]), so requiring them does not increase the order of execution time of our algorithm.

We will also make use of the following notation. As the vertex v is reached in a left-to-right processing of the endnode list, let Tv be the subtree induced by v and all of its descendants. (Note that all of the descendants of v have already been processed since they precede v in EL.) Let u be the parent of v (which is determined using PA) and let Tu' be the subtree induced by u, the children of u that precede v in EL, and the descendants of all such children. Finally, let Tu be composed of Tu', Tv and the edge (u, v). Note that Tu does not necessarily contain all of the descendants of u since there may be children of u which appear after v in EL. See Fig. 5.

The first three parameters we are interested in will be used to ensure that we

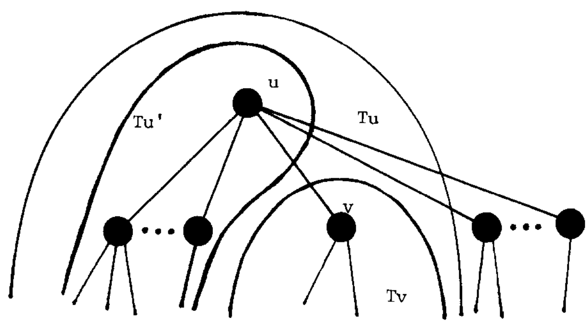

Fig. 5. Illustration of Tv, Tu, and Tu' notation.

get *minimum* dominating sets. Note that $\gamma(T)$ can be achieved without the use of these parameters, but we employ them here because they will be used in evaluating the minimum size of the intersection of two MDS's later in this section. For a vertex $u \in V(T)$, define

$$\gamma_y(Tu) = \text{MIN}\{|D|: u \in D,\ D \text{ dominates } Tu\},$$
$$\gamma_n(Tu) = \text{MIN}\{|D|: u \notin D,\ D \text{ dominates } Tu\},$$
$$\gamma_{\bar{n}}(Tu) = \text{MIN}\{|D|: u \notin D,\ D \text{ dominates } Tu - u\}.$$

That is, let $\gamma_y(Tu)$ be the minimum order of a dominating set of Tu which contains u, let $\gamma_n(Tu)$ be the minimum order of a dominating set of Tu which does not contain u, and let $\gamma_{\bar{n}}(Tu)$ be the minimum order of a dominating set of $Tu - u$ which does not contain u. This third parameter will be useful when u is to be dominated by its parent or by an as yet unprocessed child.

Note that, since any dominating set of Tu that does not contain u is also a dominating set of $Tu - u$, we have that $\gamma_{\bar{n}}(Tu) \leq \gamma_n(Tu)$ for all $u \in V(T)$. Also $\gamma(Tu)$, the minimum number of vertices in a dominating set of Tu, can be expressed by

$$\gamma(Tu) = \text{MIN}\{\gamma_y(Tu),\ \gamma_n(Tu)\}. \tag{1}$$

Now, if D is a minimum dominating set of Tu and if $u \in D$ then we may write $D = U \cup V$ where U is a minimum dominating set of Tu', $u \in U$, and V is a smallest possible vertex subset of Tv that dominates $Tv - v$ (and may or may not dominate v, since v is dominated by u). The vertex v may or may not be an element of V. Thus $\gamma_y(Tu) = \gamma_y(Tu') + \text{MIN}\{\gamma_y(Tv),\ \gamma_n(Tv),\ \gamma_{\bar{n}}(Tv)\}$. But since $\gamma_{\bar{n}}(Tv) \leq \gamma_n(Tv)$ we may write this recursive relation as

$$\gamma_y(Tu) = \gamma_y(Tu') + \text{MIN}\{\gamma_y(Tv),\ \gamma_{\bar{n}}(Tv)\}. \tag{2}$$

Similarly it is straightforward to derive the following:

$$\gamma_n(Tu) = \text{MIN}\{\gamma_n(Tu') + \gamma(Tv),\ \gamma_{\bar{n}}(Tu') + \gamma_y(Tv)\} \tag{3}$$

and

$$\gamma_{\bar{n}}(Tu) = \gamma_{\bar{n}}(Tu') + \gamma(Tv). \tag{4}$$

To see how these parameters should be initialized, consider a subtree consisting of a single vertex v. Then the minimum number of vertices needed to dominate the subtree using v is one; it is not possible to dominate the subtree without using v; and zero vertices are required to dominate the subtree minus v. Thus for any endpoint v_i of a tree T we may initialize $\gamma_y(Tv_i) = 1$, $\gamma_n(Tv_i) = \infty$, and $\gamma_{\bar{n}}(Tv_i) = 0$. And for any internal node u_i of T we may initialize $\gamma_y(Tu_i') = 1$, $\gamma_n(Tu_i') = \infty$, and $\gamma_{\bar{n}}(Tu_i') = 0$. After this initialization we may proceed through the endnode list evaluating equations (1) through (4) for u_i where u_i is the parent of the current endnode list entry v_i. Once v_n is reached, we can determine $\gamma(T) = \gamma(Tv_n) = \text{MIN}\{\gamma_y(Tv_n),\ \gamma_n(Tv_n)\}$.

For example, $\gamma(T) = 4$ for the tree of Fig. 6.

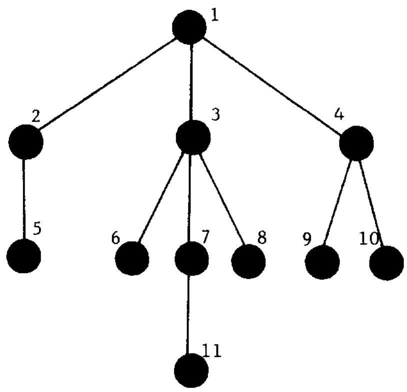

Fig. 6. A tree with $\gamma(T) = 4$.

Table 1

EL v	$\gamma_y(Tv)$	$\gamma_n(Tv)$	$\gamma_{\bar{n}}(Tv)$
5	1	∞	0
6	1	∞	0
11	1	∞	0
7	1	1	1
8	1	∞	0
9	1	∞	0
10	1	∞	0
2	1	1	1
3	2	3	3
4	1	2	2
1	5	4	4

While the three γ-type parameters were used for maintaining information about any one dominating set, we will now introduce six more parameters for maintaining information relating two dominating sets. More specifically, they are used for maintaining the minimum order of the intersection of two minimum dominating sets with certain additional properties. For $u \in V(T)$, define

$$\lambda_{yy}(Tu) = \text{MIN}\{|D_1 \cap D_2|: u \in D_1,\ u \in D_2,\ D_1 \text{ and } D_2 \text{ each dominate } Tu,\ |D_i| = \gamma_y(Tu)\};$$

$$\lambda_{nn}(Tu) = \text{MIN}\{|D_1 \cap D_2|: u \notin D_i,\ D_i \text{ dominates } Tu,\ |D_i| = \gamma_n(Tu)\};$$

$$\lambda_{\bar{n}\bar{n}}(Tu) = \text{MIN}\{|D_1 \cap D_2|: u \notin D_i,\ D_i \text{ dominates } Tu - u,\ |D_i| = \gamma_{\bar{n}}(Tu)\};$$

$$\lambda_{yn}(Tu) = \text{MIN}\{|D_1 \cap D_2|: u \in D_1,\ u \notin D_2,\ D_i \text{ dominates } Tu,\ |D_1| = \gamma_y(Tu),\ |D_2| = \gamma_n(Tu)\};$$

$$\lambda_{y\bar{n}}(Tu) = \text{MIN}\{|D_1 \cap D_2|: u \in D_1,\ u \notin D_2,\ D_1 \text{ dominates } Tu,\ |D_1| = \gamma_y(Tu),\ D_2 \text{ dominates } Tu - u,\ |D_2| = \gamma_{\bar{n}}(Tu)\};$$

$$\lambda_{n\bar{n}}(Tu) = \text{MIN}\{|D_1 \cap D_2|: u \notin D_i,\ D_1 \text{ dominates } Tu,\ D_2 \text{ dominates } Tu - u,\ |D_1| = \gamma_n(Tu),\ |D_2| = \gamma_{\bar{n}}(Tu)\}.$$

That is, $\lambda_{yy}(Tu)$ is the minimum cardinality of the intersection of two MDS's of Tu, each of which contains u; $\lambda_{nn}(Tu)$ is the minimum cardinality of the intersection of two MDS's of Tu, neither of which contains u; $\lambda_{\bar{n}\bar{n}}$ is the minimum cardinality of the intersection of two MDS's of $Tu - u$, neither of which contains u; λ_{yn} is the minimum cardinality of the intersection of two MDS's of Tu, one containing u and the other not containing u; $\lambda_{y\bar{n}}$ is the minimum cardinality of the intersection of an MDS of Tu containing u and an MDS of $Tu - u$ not containing u; and $\lambda_{n\bar{n}}$ is the minimum cardinality of the intersection of an MDS of Tu and an MDS of $Tu - u$, neither of which contains u.

To determine the initial conditions for the λ parameters, again consider a subtree consisting only of the vertex v. Any MDS of the subtree consists of exactly the vertex v, so $\lambda_{yy}(Tv) = 1$. It is not possible to dominate the subtree without using v, so $\lambda_{nn}(Tu) = \lambda_{yn}(Tv) = \lambda_{n\bar{n}}(Tv) = \infty$. No vertices are necessary to dominate $Tv - v$, so $\lambda_{\bar{n}\bar{n}}(Tv) = 0$. Since $\{v\}$ is a MDS of Tv and $\emptyset$ is a MDS of $Tv - v$, $\lambda_{y\bar{n}}(Tv) = 0$. We therefore initialize all endpoints of a tree T in this manner. Also, for internal nodes u, $\lambda_{yy}(Tu') = 1$, $\lambda_{nn}(Tu') = \lambda_{yn}(Tu') = \lambda_{n\bar{n}}(Tu') = \infty$, and $\lambda_{\bar{n}\bar{n}}(Tu') = \lambda_{y\bar{n}}(Tu') = 0$.

After this initialization, proceed through EL starting with v_1 and evaluating each of equations (5) through (10) for u_i where v_i is the current element of EL. Once the vertex v_n is reached we are ready to determine M_γ. First, since $\lambda_{y\bar{n}}(Tv_n)$, $\lambda_{n\bar{n}}(Tv_n)$, and $\lambda_{\bar{n}\bar{n}}(Tv_n)$ consider sets which are only required to dominate $Tv_n - v_n$, they are not used in choosing M_γ. Second, we must insure that only minimum dominating sets of $T = Tv$ are considered, so if $\gamma_y(Tv_n) < \gamma_n(Tv_n)$ then $M_\gamma(T) = \lambda_{yy}(Tv_n)$; if $\gamma_y(Tv_n) > \gamma_n(Tv_n)$ then $M_\gamma(T) = \lambda_{nn}(Tv_n)$; and if $\gamma_y(Tv_n) = \gamma_n(Tv_n)$ then $M_\gamma(T) = \min\{\lambda_{yy}(Tv_n), \lambda_{yn}(Tv_n), \lambda_{nn}(Tv_n)\}$.

The γ-recurrences previously given as formulae (2), (3) and (4) appear in the following algorithm marked as lines (2), (3) and (4). The λ-recurrences appear in lines marked (5)–(10).

Verification of these recurrences is somewhat straightforward. We explain only one of these, namely λ_{yn} (line (8) in the algorithm). In the algorithm, we write $\lambda_{yn}(u)$ for $\lambda_{yn}(Tu)$, $\gamma_n(u)$ for $\gamma_n(Tu)$, etc.

The parameter $\lambda_{yn}(u)$ represents the minimum cardinality of the intersection of γ_y-set of Tu (that is, a set D_1 which dominates Tu, contains u, and has cardinality $\gamma_y(Tu)$) and a γ_n-set of Tu (that is, a set D_2 which dominates Tu, does not contain u, and has cardinality $\gamma_n(Tu)$). To formulate the appropriate recurrences, we consider the intersection of such sets D_1 and D_2 with the vertices of Tv, specifically with the vertex v and its immediate children.

This leads to six possibilities:

(1) We first suppose $v \in D_1$ and $v \in D_2$. This can only happen if $\gamma_y(Tu) = \gamma_y(Tu') + \gamma_y(Tv)$ and $\gamma_n(Tu) = \gamma_{\bar{n}}(Tu') + \gamma_y(Tv)$. In such a case, $\lambda_{yn}(Tu) \leqslant \lambda_{y\bar{n}}(Tu') + \lambda_{yy}(Tv)$. Thus in the algorithm, if the two γ conditions are met we set a variable $D1$ equal to $\lambda_{yy}(Tv) + \lambda_{y\bar{n}}(Tu')$. If not both γ conditions are met then this case must be excluded, so we set $D1 := \infty$. Variables $D2$ through $D6$ are similarly defined in cases (2) through (6) below. Noting that $D3$ always equals ∞, we then set $\lambda_{yn}(Tu) := \mathrm{MIN}(D1, D2, D4, D5, D6)$ in line (8) of the algorithm.

(2) Suppose v is not an element of D_1 or D_2 but v is dominated by a vertex in each of D_1 and D_2 other than by u. In this case, $D2$ equals $\lambda_{nn}(Tv) + \lambda_{yn}(Tu')$ or $D2$ equals ∞.

(3) Suppose v is not an element of D_1 or D_2 and is not necessarily dominated by a vertex other than u in each set. These conditions will not quarantee that v is dominated, thus $D3$ would always be infinity.

(4) Suppose v is an element of D_1 and v is not an element of D_2 but a child of v

is an element of D_2. In this case, $D4 = \lambda_{yn}(Tu') + \lambda_{yn}(Tv)$ or $D4 = \infty$. (If the role of v is reversed in the two sets, the situation will be subsumed in case (5).)

(5) Suppose v is not an element of D_1 (and neither are any of its children) and v is an element of D_2. In this case, $D5 = \lambda_{y\bar{n}}(Tu') + \lambda_{y\bar{n}}(Tv)$ or $D5 = \infty$. (If the role of v is reversed in the two sets, the situation is excluded since the conditions will not guarantee that v is dominated.)

(6) Suppose v is not an element of D_1 or D_2 and that v is not necessarily dominated by a vertex other than u in D_1. In this case, $D6 = \lambda_{\bar{n}n}(Tv) + \lambda_{yn}(Tu')$ or $D6 = \infty$. (If the role of v is reversed in the two sets, the situation is excluded since the conditions will not quarantee that v is dominated in D_2.)

The other λ recurrences can be similarly justified and the algorithm follows.

Algorithm MGAMMA (T) (* for finding the minimum cardinality of the intersection of two minimum dominating sets for a tree *)

For i: =1 **to** n **do**
  $\gamma_y(i) := 1$; $\gamma_n(i) := \infty$; $\gamma_{\bar{n}}(i) := 0$; $\gamma(i) := 1$;
  $\lambda_{yy}(i) := 1$; $\lambda_{nn}(i) := \infty$; $\lambda_{\bar{n}\bar{n}}(i) := 0$;
  $\lambda_{yn}(i) := \infty$; $\lambda_{y\bar{n}}(i) := 0$; $\lambda_{n\bar{n}}(i) := \infty$;

For i: =1 **to** n **do**
  γ_y New $:= \gamma_y(\text{PA}(i)) + \text{MIN}(\gamma_y(\text{EL}(i)), \gamma_{\bar{n}}(\text{EL}(i)))$ (2)
  γ_n New $:= \text{MIN}(\gamma_n(\text{PA}(i)) + \gamma(\text{EL}(i)), \gamma_{\bar{n}}(\text{PA}(i)) + \gamma_y(\text{EL}(i)))$ (3)
  $\gamma_{\bar{n}}$ New $:= \gamma_{\bar{n}}(\text{PA}(i)) + \gamma(\text{EL}(i))$ (4)

(* COMPUTING λ_{yy} *)
  If γ_y New $= \gamma_y(\text{PA}(i)) + \gamma_y(\text{EL}(i))$
    then $A1 := \lambda_{yy}(\text{EL}(i))$
    else $A1 := \infty$
  If γ_y New $= \gamma_y(\text{PA}(i)) + \gamma_{\bar{n}}(\text{EL}(i))$
    then $A3 := \lambda_{\bar{n}\bar{n}}(\text{EL}(i))$
    else $A3 := \infty$

  If γ_y New $= \gamma_y(\text{PA}(i)) + \gamma_y(\text{EL}(i))$ **and** γ_y New $= \gamma_y(\text{PA}(i)) + \gamma_{\bar{n}}(\text{EL}(i))$
    then $A5 := \lambda_{y\bar{n}}(\text{EL}(i))$
    else $A5 := \infty$
    $\lambda_{yy}(\text{PA}(i)) := \lambda_{yy}(\text{PA}(i)) + \text{MIN}(A1, A3, A5)$ (5)

(* COMPUTING λ_{nn} *)
  If γ_n New $= \gamma_{\bar{n}}(\text{PA}(i)) + \gamma_y(\text{EL}(i))$
    then $B1 := \lambda_{\bar{n}\bar{n}}(\text{PA}(i)) + \lambda_{yy}(\text{EL}(i))$
    else $B1 := \infty$

If γ_n New $= \gamma_n(\text{PA}(i)) + \gamma_n(\text{EL}(i))$
then $B2 := \lambda_{nn}(\text{PA}(i)) + \lambda_{nn}(\text{EL}(i))$
else $B2 := \infty$

If γ_n New $= \gamma_{\bar{n}}(\text{PA}(i)) + \gamma_y(\text{EL}(i))$ **and** γ_n New $= \gamma_n(\text{PA}(i)) + \gamma_n(\text{EL}(i))$
then $B4 := \lambda_{n\bar{n}}(\text{PA}(i)) + \lambda_{yn}(\text{EL}(i))$
else $B4 := \infty$
$\lambda_{nn}(\text{PA}(i)) := \text{MIN}(B1, B2, B4)$ (6)

(* COMPUTING $\lambda_{\bar{n}\bar{n}}$ *)
If $\gamma_{\bar{n}}$ New $= \gamma_{y\bar{n}}(\text{PA}(i)) + \gamma_y(\text{EL}(i))$
then $C1 := \lambda_{yy}(\text{EL}(i))$
else $C1 := \infty$

If $\gamma_{\bar{n}}$ New $= \gamma_{\bar{n}}(\text{PA}(i)) + \gamma_n(\text{EL}(i))$
then $C2 := \lambda_{nn}(\text{EL}(i))$
else $C2 := \infty$

If $\gamma_{\bar{n}}$ New $= \gamma_{\bar{n}}(\text{PA}(i)) + \gamma_y(\text{EL}(i))$ **and** $\gamma_{\bar{n}}$ New $= \gamma_{\bar{n}}(\text{PA}(i)) + \gamma_n(\text{EL}(i))$
then $C4 := \lambda_{yn}(\text{EL}(i))$
else $C4 := \infty$
$\lambda_{\bar{n}\bar{n}}(\text{PA}(i)) := \lambda_{\bar{n}\bar{n}}(\text{PA}(i)) + \text{MIN}(C1, C2, C4)$ (7)

(* COMPUTING λ_{yn} *)
If γ_y New $= \gamma_y(\text{PA}(i)) + \gamma_y(\text{EL}(i))$ **and** λ_n New $= \gamma_{\bar{n}}(\text{PA}(i)) + \gamma_y(\text{EL}(i))$
then $D1 := \lambda_{yy}(\text{EL}(i)) + \lambda_{y\bar{n}}(\text{PA}(i))$
else $D1 := \infty$

If γ_y New $= \gamma_y(\text{PA}(i)) + \gamma_n(\text{EL}(i))$ **and** γ_n New $= \gamma_n(\text{PA}(i)) + \gamma_n(\text{EL}(i))$
then $D2 := \lambda_{nn}(\text{EL}(i)) + \lambda_{yn}(\text{PA}(i))$
else $D2 := \infty$

If γ_y New $= \gamma_y(\text{PA}(i)) + \gamma_y(\text{EL}(i))$ **and** γ_n New $= \gamma_n(\text{PA}(i)) + \gamma_n(\text{EL}(i))$
then $D4 := \lambda_{yn}(\text{PA}(i)) + \lambda_{yn}(\text{EL}(i))$
else $D4 := \infty$

If γ_y New $= \gamma_y(\text{PA}(i)) + \gamma_{\bar{n}}(\text{EL}(i))$ **and** γ_n New $= \gamma_{\bar{n}}(\text{PA}(i)) + \gamma_y(\text{EL}(i))$
then $D5 := \lambda_{y\bar{n}}(\text{PA}(i)) + \lambda_{y\bar{n}}(\text{EL}(i))$
else $D5 := \infty$

If γ_y New $= \gamma_y(\text{PA}(i)) + \gamma_{\bar{n}}(\text{EL}(i))$ **and** γ_n New $= \gamma_n(\text{PA}(i)) + \gamma_n(\text{EL}(i))$
then $D6 := \lambda_{\bar{n}n}(\text{EL}(i)) + \lambda_{yn}(\text{PA}(i))$
else $D6 := \infty$
$\lambda_{yn}(\text{PA}(i)) := \text{MIN}(D1, D2, D4, D5, D6)$ (8)

(* COMPUTING $\lambda_{y\bar{n}}$ *)
If γ_y New $= \gamma_y(\text{PA}(i)) + \gamma_y(\text{EL}(i))$ **and** $\gamma_{\bar{n}}$ New $= \gamma_{\bar{n}}(\text{PA}(i)) + \gamma_y(\text{EL}(i))$
then $E1 := \lambda_{yy}(\text{EL}(i))$
else $E1 := \infty$

If γ_y New $= \gamma_y(\mathrm{PA}(i)) + \gamma_n(\mathrm{EL}(i))$ **and** $\gamma_{\bar{n}}$ New $= \gamma_{\bar{n}}(\mathrm{PA}(i)) + \gamma_n(\mathrm{EL}(i))$
  then $E2 := \lambda_{nn}(\mathrm{EL}(i))$
  ELSE $E2 := \infty$

If γ_n New $= \gamma_y(\mathrm{PA}(i)) + \gamma_y(\mathrm{EL}(i))$ **and** $\gamma_{\bar{n}}$ New $= \gamma_{\bar{n}}(\mathrm{PA}(i)) + \gamma_n(\mathrm{EL}(i))$
  then $E4 := \lambda_{yn}(\mathrm{EL}(i))$
  else $E4 := \infty$

If γ_y New $= \gamma_y(\mathrm{PA}(i)) + \gamma_{\bar{n}}(\mathrm{EL}(i))$ **and** $\gamma_{\bar{n}}$ New $= \gamma_{\bar{n}}(\mathrm{PA}(i)) + \gamma_y(\mathrm{EL}(i))$
  then $E5 := \lambda_{y\bar{n}}(\mathrm{EL}(i))$
  else $E5 := \infty$

If γ_y New $= \gamma_y(\mathrm{PA}(i)) + \gamma_{\bar{n}}(\mathrm{EL}(i))$ **and** $\gamma_{\bar{n}}$ New $= \gamma_{\bar{n}}(\mathrm{PA}(i)) + \gamma_n(\mathrm{EL}(i))$
  then $E6 := \lambda_{n\bar{n}}(\mathrm{EL}(i))$
  else $E6 := \infty$
  $\lambda_{y\bar{n}}(\mathrm{PA}(i)) = \lambda_{y\bar{n}}(\mathrm{PA}(i)) + \mathrm{MIN}(E1, E2, E4, E5, E6)$ (9)

(* COMPUTING $\lambda_{n\bar{n}}$ *)
If γ_n New $= \gamma_{\bar{n}}(\mathrm{PA}(i)) + \gamma_y(\mathrm{EL}(i))$ **and** $\gamma_{\bar{n}}$ New $= \gamma_{\bar{n}}(\mathrm{PA}(i)) + \gamma_y(\mathrm{EL}(i))$
  then $F1 := \lambda_{yy}(\mathrm{EL}(i)) + \lambda_{\bar{n}\bar{n}}(\mathrm{PA}(i))$
  else $F1 := \infty$

If γ_n New $= \gamma_n(\mathrm{PA}(i)) + \gamma_n(\mathrm{EL}(i))$ **and** $\gamma_{\bar{n}}$ New $= \gamma_{\bar{n}}(\mathrm{PA}(i)) + \gamma_n(\mathrm{EL}(i))$
  then $F2 := \lambda_{nn}(\mathrm{EL}(i)) + \lambda_{n\bar{n}}(\mathrm{PA}(i))$
  else $F2 := \infty$

If γ_n New $= \gamma_{\bar{n}}(\mathrm{PA}(i)) + \gamma_y(\mathrm{EL}(i))$ **and** $\gamma_{\bar{n}}$ New $= \gamma_{\bar{n}}(\mathrm{PA}(i)) + \gamma_n(\mathrm{EL}(i))$
  then $F4 := \lambda_{yn}(\mathrm{EL}(i)) + \lambda_{\bar{n}\bar{n}}(\mathrm{PA}(i))$
  else $F4 := \infty$

If γ_n New $= \gamma_n(\mathrm{PA}(i)) + \gamma_n(\mathrm{EL}(i))$ **and** $\gamma_{\bar{n}}$ New $= \gamma_{\bar{n}}(\mathrm{PA}(i)) + \gamma_y(\mathrm{EL}(i))$
  then $F4B := \lambda_{yn}(\mathrm{EL}(i)) + \lambda_{n\bar{n}}(\mathrm{PA}(i))$
  else $F4B := \infty$
  $\lambda_{n\bar{n}}(\mathrm{PA}(i)) := \mathrm{MIN}(F1, F2, F4, F4B)$ (10)

$\gamma_y(\mathrm{PA}(i)) := \gamma_y\mathrm{New};$
$\gamma_n(\mathrm{PA}(i)) := \gamma_n\mathrm{New};$
$\gamma_{\bar{n}}(\mathrm{PA}(i)) := \gamma_{\bar{n}}\mathrm{New};$
{end for loop}

(* CONCLUDING *)
$\gamma := \mathrm{MIN}(\gamma_y(\mathrm{EL}(n)), \gamma_n(\mathrm{EL}(n)));$
If $\gamma_y(\mathrm{EL}(n)) < \gamma_n(\mathrm{EL}(n))$
  then $M_\gamma := \lambda_{yy}(\mathrm{EL}(n));$

If $\gamma_y(\mathrm{EL}(n)) > \gamma_n(\mathrm{EL}(n))$
  then $M_\gamma := \lambda_{nn}(\mathrm{EL}(n));$

If $\gamma_y(\mathrm{EL}(n)) = \gamma_n(\mathrm{EL}(n))$
  then $M_\gamma := \min\{\lambda_{yy}(\mathrm{EL}(n)), \lambda_{yn}(\mathrm{EL}(n)), \lambda_{nn}(\mathrm{EL}(n))\};$

Table 2. Parameters λ for tree of Fig. 6 showing that $M_{\gamma}(T) = \text{MIN}\{3, 2, 2\} = 2$

EL v	$\lambda_{yy}(Tv)$	$\lambda_{nn}(Tv)$	$\lambda_{\bar{n}\bar{n}}(Tv)$	$\lambda_{yn}(Tv)$	$\lambda_{y\bar{n}}(Tv)$	$\lambda_{n\bar{n}}(Tv)$
5	1	∞	0	∞	0	∞
6	1	∞	0	∞	0	∞
11	1	∞	0	∞	0	∞
7	1	1	1	0	0	1
8	1	∞	0	∞	0	∞
9	1	∞	0	∞	0	∞
10	1	∞	0	∞	0	∞
2	1	1	1	0	0	1
3	1	2	2	0	0	2
4	1	2	2	0	0	2
1	3	2	2	2	2	2

For the tree of Fig. 6, Table 2 represents the values of all of the λ parameters and hence $M_{\gamma}(T) = 2$.

4. Further extensions

The algorithm presented in Section 3 determines the parameter $M_{\gamma}(T)$, the minimum cardinality of the intersection of two MDS's of tree T. Two minimum dominating sets whose intersection has such cardinality can be obtained by one additional scan (this time right-to-left rather than left-to-right) of the endpoint list. (See Grinstead [20].) We will not fully present the procedure here, but only mention that in order to find the sets some additional information is collected in the first scan. For example, in the right-to-left scan of the endpoint list, the root of the tree is the first to be processed. As each vertex is reached, we will decide whether or not to use the vertex in $D1$ (the first MDS) and whether or not to use it in $D2$ (the second MDS). If, in either set, it is ever decided not to use a vertex which has not already been dominated by its parent, we must have some information about its children as to which one will 'cost' the least in terms of the size of the MDS and in terms of the cardinality of the intersection of $D1$ and $D2$.

A more complicated, but still linear, algorithm for finding the minimum cardinality of the intersection of two minimum dominating sets in a series-parallel graph will appear in Grinstead [20]. Since series parallel graphs have two terminals, this algorithm must consider nine γ-type parameters (rather than three for the tree case) and forty-five λ-type parameters (as opposed to six for the tree case). Since there are three basic ways of connecting two series-parallel subgraphs, each of the nine γ-type and forty-five λ-type parameters must have three subcases.

Finally, as previously noted, we view this minimum intersection MDS problem as a prototype of many possible problems involving such sets as dominating and/or independent vertex sets, with details concerning various problems to appear in Grinstead [20].

5. Addendum

As indicated in the introduction, much work is being done in the developing theory of polynomial/linear algorithms for graph theoretic problems. Some quite recent work is concerned with predicting the nature of problems for which there will exist linear time algorithms on recursive families of graphs. Such work includes that of Bern, Lawler and Wong [7], Bodlaender [10], Borie, Parker and Tovey [11], Mahajan and Peters [27], and Seese [31].

We also note that results on $M_\gamma(G)$ for series–parallel graphs appear in Grinstead and Slater [21].

References

[1] S. Arnborg and A. Proskurowski, Linear-time algorithms for NP-hard problems restricted to partial k-trees, Report No. TRITA-NA-8404, The Royal Insitute of Technology, Sweden, 1984.

[2] D.W. Bange, A.E. Barkauskas and P.J. Slater, A constructive characterization of trees with two disjoint minimum dominating sets, in: Proceedings Ninth S.E. Conference on Combinatorics, Graph Theory and Computing (Utilitas Mathematica, Winnipeg, 1978) 101–112.

[3] D.W. Bange, A.E. Barkauskas and P.J. Slater, Disjoint dominating sets in trees, Sandia Laboratories Report, SAND 78-1087J, 1978.

[4] D.W. Bange, A.E. Barkauskas and P.J. Slater, Efficient dominating sets in graphs, in: R.D. Ringeisen, and F.S. Roberts, eds., Applications of Discrete Mathematics (SIAM, Philadelphia, PA, 1988) 189–199.

[5] D.W. Bange, A.E. Barkauskas and P.J. Slater, Disjoint domination algorithms for trees, presented at the First Clemson University/ONR Mini-Conference on Discrete Mathematics, October, 1986.

[6] M.W. Bern, E.L. Lawler and A.L. Wong, Why certain subgraph computations require only linear time, in: Proceedings 26th Annual Symposium on the Foundations of Computer Science (Portland, OR, 1985) 117–125.

[7] M.W. Bern, E.L. Lawler and A.L. Wong, Linear time computation of optimal subgraphs of decomposable graphs, J. Algorithms 8 (1987) 216–235.

[8] T. Beyer, Proskurowski, S. Hedetniemi and S. Mitchell, Independent domination in trees, in: Proceedings Eight S.E. Conference on Combinatorics, Graph Theory and Computing (Utilitas Mathematica, Winnipeg, 1977) 231–328.

[9] H.L. Bodlaender, Polynomial algorithms for chromatic index and graph isomorphism on partial k-trees, Tech. Rept. RUU-CS-87-17, Dept. of Computer Science, Univ. of Utrecht, Netherlands, October 1987.

[10] H.L. Bodlaender, Dynamic programming on graphs with bounded tree-width, Ph.D. Thesis, MIT/LCS/TR-394, Massachusetts Institute of Technology, June 1987; and Tech. Rept. RUU-CS-87-22, Dept. Computer Science, Univ. Utrecht, Netherlands, November 1987.

[11] R.B. Borie, R. Gary Parker and Craig A. Tovey, Automatic generation of linear algorithms from predicate calculus descriptions of problems on recursively constructed graph families, preprint, Georgia Institute of Technology, July 1988.

[12] E.J. Cockayne, S. Goodman and S.T. Hedetniemi, A linear algorithm for the domination number of a tree, Inform. Process. Lett. 4 (1975) 41–44.

[13] E.J. Cockayne and S.T. Hedetniemi, Optimal domination in graphs, IEEE Trans. Circuits and Systems CAS-22 (1975) 855–857.

[14] G.J. Chang and G.L. Nemhauser, R-domination on block graphs, Oper. Res. Lett. 1 (1982) 214–218.

[15] C.J. Colbourn, P.J. Slater and L.K. Stewart, Locating-dominating sets in seriesparallel

networks, Proceedings 16th Annual Conference on Numerical Mathematics and Computing, Winnipeg, 1986, Congr. Numer. 56 (1987) 135–162.
[16] A.K. Dewdney, Fast Turing reductions between problems in NP, Chapter 4: Reductions between NP-complete problems, Report #71, Dept. Computer Science, Univ. Western Ontario, 1981.
[17] M.R. Garey, D.S. Johnson and L. Stockmeyer, Some simplified NP-complete graph problems, Theoret. Comput. Sci. 1 (1976) 237–267.
[18] M.R. Garey, D.S. Johnson and R.E. Tarjan, 1976, unpublished.
[19] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness (Freeman, New York, 1979).
[20] D.L. Grinstead, Algoritmic templates and multiset problems in graphs, Ph.D Thesis, Univ. of Alabama in Huntsville, 1989.
[21] D.L. Grinstead and P.J. Slater, On the minimum intersection of minimum dominating sets in seris-parallel graphs, 1988, submitted for publication.
[22] D.S. Johnson, The NP-completeness column: An ongoing guide, J. Algorithms 6 (1985) 434–451.
[23] R.M. Karp, Reducibility among combinatorial problems, in: R.E. Miller and J.W. Thatcher, eds, Complexity of Computer Computations (Plenum, New York, 1972) 85–103.
[24] T. Kikuno, N. Yoshida and Y. Kakuda, A linear algorithm for the domination number of a series-parallel graph, Discrete Appl. Math. 5 (1983) 299–311.
[25] D.E. Knuth, The Art of Computer Programming, Vol. 1: Fundamental Algorithms (Addison-Wesley, Reading, MA, 1968) 334–338.
[26] E.L. Lawler and P.J. Slater, A linear time algorithm for finding an optimal dominating subforest of a tree, in: Graph Theory with Applications to Algorithms and Computer Science, Kalamazoo, MI, 1984 (Wiley, New York, 1985) 501–506.
[27] S. Mahajan and J.G. Peters, Algorithms for regular properties in recursive graphs, in: Twenty-fifth Annual Allerton Conference on Communications, Control, and Computing, 1987, 14–23.
[28] J. Pfaff, R. Laskar and S.T. Hedetniemi, Linear algorithms for independent domination and total domination in seriesparallel graphs, Congr. Numer. 45 (1985) 71–82.
[29] N. Robertson and P.D. Seymour, Graph minors II: Algorithmic aspects of tree-width, J. Algorithms 7 (1986) 309–322.
[30] T.J. Schaefer, The complexity of satisfiability problems, in: Proceedings 10th Annual ACM Synposium on Theory of Computing (Association for Computing Machinery, New York, 1978) 216–226.
[31] D. Seese, Tree-partite graphs and the complexity of algorithms, Tech. Rept. P-MATH-08/86, Akademie der Wissenschaften der DDR, Karl-Weierstrass-Institut fur Mathematik, Berlin, 1986.
[32] P.J. Slater, R-domination in graphs, J. Assoc. Comput. Mach. 23 (1976) 446–450.
[33] K. Takamizawa, T. Nishizeki and N. Saito, Linear-time computability of combinatorial problems on series-parallel graphs, J. Assoc. Comput. Mach. 29 (1982) 623–641.
[34] R.E. Tarjan, Depth-first search and linear graph algorithms, SIAM J. Comput. 1 (1972) 146–160.
[35] T.V. Wimer, Linear algorithms on k-terminal graphs, Ph.D. Dissertation, Computer Science Department, Clemson University, 1987.